LOUIS FRÉCON

ÉLÉMENTS DE MATHÉMATIQUES DISCRÈTES

Collection des sciences appliquées de l'INSA de Lyon

Eléments de mathématiques discrètes

Louis Frécon

PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES

TABLE DES MATIÈRES

DÉPARTEMENT D'INFORMATIQUE V
SOMMAIRE VII

Première partie **Fondements** 1

Chapitre 0 MÉMENTO DE LOGIQUE 3
0.1 Logique des propositions 3
0.2 Logique des prédicats 11
0.3 Exercices 15
0.4 Problèmes 17
0.5 Piste de réflexion 18
0.6 Lectures 18

Chapitre 1 ENSEMBLES & ÉLÉMENTS 21
1.1 Présentation 21
1.2 Inclusions 23
1.3 Parties d'un ensemble 25
1.4 Union 26
1.5 Intersection 27
1.6 Différence ensembliste 28
1.7 Couvertures 29
1.8 Différence symétrique 31
1.9 Partitions 32
1.10 Paires & produits cartésiens 34
1.11 Exercices 35
1.12 Problèmes 37
1.13 Lectures 40

Chapitre 2 RELATIONS BINAIRES 41
2.1 Vue en extension 41
2.2 Vue en compréhension 44
2.3 Relation inverse 45
2.4 Union, intersection, finesse 46
2.5 Composition 46
2.6 Exercices 47
2.7 Problèmes 48
2.8 Lectures 48

Chapitre 3 FONCTIONS 51
3.1 Relations binaires régulières 51
3.2 Fonctions 51
3.3 Surjection 53
3.4 Image & inverse 54
3.5 Injection & bijection 54
3.6 Synoptiques 56
3.7 Composition de fonctions 57
3.8 Ensembles dénombrables et finis 58
3.9 Relations n-aires 59
3.10 Fonctions généralisées 61
3.11 Exercices 63
3.12 Problèmes 64
3.13 Lectures 65

Chapitre 4 RELATIONS BINAIRES INTERNES 67
4.1 Présentation 67
4.2 Propriétés des relations binaires internes 68
4.3 Restriction 72
4.4 Equivalence 72
4.5 Ordres 75
4.6 Préordre 79
4.7 Récapitulatifs 84
4.8 Exercices 85
4.9 Problèmes 87
4.10 Thèmes de réflexion 92
4.11 Lectures 94

Chapitre 5 FONCTIONS, CALCULABILITÉ, RÉCURRENCE 95
5.1 Calculabilite, decidabilité 95
5.2 Calculabilité et récursivité 96
5.3 Récursivités convergente, divergente, stationnaire 101
5.4 Ensembles récursifs et récursivement énumérables 103
5.5 Aux sources de l'arithmétique et de l'algèbre 104
5.6 Exercices 106
5.7 Problèmes 107
5.8 Pistes de réflexion 108
5.9 Lectures 109

Chapitre 6 NOTION DE COMPLEXITÉ 111
6.1 Premier exemple 111
6.2 Règles d'estimation de la complexité 112
6.3 Préordre de complexité 115
6.4 Classes de complexité 115
6.5 Optimisation du calcul d'une fonction 117
6.6 Représentations d'une table creuse 121
6.7 Exercices 123
6.8 Problèmes 124
6.9 Lectures 125

Deuxième partie Graphes 127

Chapitre 7 DES POINTS ET DES FLÈCHES 129
7.1 Présentation 129
7.2 Degrés & demi-degrés 129
7.3 Prédécesseurs, successeurs, adjacents 130
7.4 Sommets initiaux, terminaux, isolés 130
7.5 Sous-graphe 130
7.6 Graphe partiel 131
7.7 Graphe complet, cliques 131
7.8 Représentations 131
7.9 Propriétés des relations binaires et graphes associés 133
7.10 Propriété faible / propriété forte 135
7.11 Graphe valué 135
7.12 Exercices 136
7.13 Problèmes 136
7.14 Thèmes de réflexion 137
7.15 Lectures 137

Chapitre 8 CHEMINS & CIRCUITS 139
8.1 Les chemins 139
8.2 Circuits et boucles dans un graphe 141
8.3 Chemins élémentaires dans un graphe 141
8.4 Recherche de chemins dans un graphe 143
8.5 Méthodes matricielles 145
8.6 Méthode locale 1 : recherche par niveaux 152
8.7 Méthode locale 2 : méthode du meilleur d'abord 155
8.8 Méthode locale 3 : recherche en profondeur 158
8.9 Problèmes hamiltoniens 161
8.10 Problèmes eulériens 162
8.11 Exercices 164
8.12 Problèmes 165
8.13 Thèmes de réflexion 169
8.14 Lectures 169

Chapitre 9 FERMETURES TRANSITIVES 171
9.1 Fermeture transitive et descendance stricte 171
9.2 Fermeture réflexo-transitive et descendance large 171
9.3 Interprétation des relations quelconques 172
9.4 Accessibilité dans les systèmes à états 174
9.5 Exercices 179
9.6 Problèmes 179
9.7 Lectures 181

Chapitre 10 ARBRES & CONNEXITÉ 183
10.1 Chaînes & cycles 183
10.2 Graphe (fortement) connexe 184
10.3 Composante connexe d'un graphe 184

10.4 Composantes fortement connexes 185
10.5 Centres, rayon, diamètre d'un graphe 185
10.6 Arbres, forêts, hiérarchies, arborescences 187
10.7 Arbre couvrant 188
10.8 Arbre couvrant minimal 189
10.9 Points, ensembles d'articulation 190
10.10 Isthme 191
10.11 k-Connexité et robustesse structurelle 191
10.12 Multi-graphes 192
10.13 Exercices 196
10.14 Problèmes 197
10.15 Pistes de réflexion 198
10.16 Vocabulaire 198
10.17 Lectures 200

Chapitre 11 GRAPHES MULTIPARTIS 201
11.1 Présentation 201
11.2 Tests de multipartisme 202
11.3 Jeux & fonctions de Grundy 204
11.4 Coloriage de graphes 207
11.5 Graphes bipartis 212
11.6 Graphes et hypergraphes 214
11.7 Exercices 218
11.8 Problèmes 219
11.9 Pistes de réflexion 220
11.10 Lectures 220

Troisième partie **Algèbres** 221

Chapitre 12 OPÉRATEURS & ALGÈBRES 223
12.1 Opérateurs & signatures 223
12.2 Syntaxe des expressions 224
12.3 Sémantique formelle des expressions 225
12.4 Algèbres 225
12.5 Propriétés des opérateurs 227
12.6 Eléments remarquables pour un opérateur 228
12.7 Déparenthésages 231
12.8 Propriétés inter–opérateurs 236
12.9 Morphismes 237
12.10 Formes normales & canoniques 239
12.11 Exercices 240
12.12 Problèmes 241
12.13 Pistes de réflexion 242
12.14 Lectures 243

Chapitre 13 MONOÏDES & GROUPES 245
13.1 Présentation 245
13.2 Semi-Groupe 248
13.3 Groupe 249

13.4 Monoïde finiment engendré 251
13.5 Langages formels 254
13.6 Grammaires 254
13.7 Exercices 256
13.8 Problèmes 258
13.9 Pistes de réflexion 260
13.10 Lectures 261

Chapitre 14 DIOÏDES 263
14.1 Présentation 263
14.2 Semi-anneau 264
14.3 Anneau 264
14.4 Corps 265
14.5 Graphe valué 265
14.6 Calculs matriciels 267
14.7 Thèse centrale 267
14.8 Taille illimitée 267
14.9 Taille limitée 269
14.10 Exemples 270
14.11 Graphes multivalués 273
14.12 Exercices 274
14.13 Problèmes 277
14.14 Pistes de réflexion 277
14.15 Lectures 279
14.16 Famille des dioïdes 279

Chapitre 15 ALGÈBRE DE BOOLE 281
15.1 Intérêt 281
15.2 Algèbre de boole élémentaire 282
15.3 Modélisation booléenne de technologies 283
15.4 Fonctions booléennes 285
15.5 Géométrie booléenne 286
15.6 Bases d'une fonction 289
15.7 Des sommes de monômes aux produits de clauses 300
15.8 Approches dichotomiques 305
15.9 Equations booléennes 308
15.10 Anneau de Boole 309
15.11 Equivalences majeures 312
15.12 Exercices 313
15.13 Problèmes 313
15.14 Piste de réflexion 314
15.15 Lectures 314

Chapitre 16 ALGÈBRE DE KLEENE & AUTOMATES À ÉTATS FINIS 317
16.1 Notion d'automate à états finis 317
16.2 Grammaires de Kleene 321
16.3 Algèbre de Kleene 322
16.4 Propriété centrale 325

16.5 Des expressions régulières aux graphes de transition....328
16.6 Synoptique....333
16.7 Exercices....333
16.8 Problèmes....335
16.9 Pistes de réflexion....336
16.10 Lectures....337

Annexe INDICATIONS SUR LES EXERCICES & PROBLÈMES....339
Chapitre 0 : Memento de logique....339
Chapitre 1 : Ensembles & éléments....341
Chapitre 2 : Relations binaires....344
Chapitre 3 : Fonctions....345
Chapitre 4 : Relations binaires internes....348
Chapitre 5 : Fonctions, calculabilité, récurrence....352
Chapitre 6 : Notion de complexité....354
Chapitre 7 : Des points et des flèches....355
Chapitre 8 : Chemins & circuits....356
Chapitre 9 : Fermetures transitives....357
Chapitre 10 : Arbres & connexité....359
Chapitre 11 : Graphes multipartis....359
Chapitre 12 : Opérateurs et algèbres....361
Chapitre 13 : Monoïdes....361
Chapitre 14 : Dioïdes....363
Chapitre 15 : Algèbre de Boole....364
Chapitre 16 : Algèbres de Kleene & automates à états finis....364

INDEX....367

DÉPARTEMENT D'INFORMATIQUE

Le Département d'Informatique de l'INSA de Lyon, fondé en 1969, a été l'une des premières écoles d'ingénieurs en informatique, créée en France. Ce département forme 120 diplômés par promotion.

Les élèves ingénieurs acquièrent une formation de *généraliste en informatique* leur permettant de maîtriser à la fois la diversité des techniques informatiques (architecture, génie logiciel, systèmes d'information communicants, intelligence artificielle, systèmes temps réel, architectures multi-tâches et gestion du parallélisme, réseau et téléinformatique...) et la variété des domaines d'application (applications dites de gestion, mise en œuvre d'ERP, systèmes de production, systèmes industriels, intégration de la dimension multimédia, ingénierie des connaissances...). Ces domaines, aussi bien au niveau technique qu'applicatif, évoluent en permanence depuis la naissance de nouvelles techniques (appréhendées grâce à une veille technologique), à leur déclin, puis à leur disparition ; leur développement est conditionné conjointement par leur maturité et celle du domaine auquel on les applique (intranets, systèmes embarqués...). La formation des élèves ingénieurs du département est en permanence adaptée à ce contexte évolutif. Elle est complétée par des stages en entreprise constituant des ouvertures vers les futurs métiers.

Le contenu de ce livre correspond à un enseignement dispensé durant la formation des élèves ingénieurs informaticiens du département. Il se propose de couvrir les mathématiques discrètes en se concentrant sur ce qui est utilisé dans les divers domaines de la *science informatique*. Le choix des matières abordées est adéquat et convaincant. Il se situe à la convergence entre la théorie et les applications. Ce livre sera un outil précieux non seulement pour les élèves du Département Informatique, mais aussi pour les ingénieurs et les informaticiens souhaitant parfaire leurs connaissances. Les exercices, problèmes et pistes de réflexion proposés en fin de chapitre sont à la fois très originaux et d'un grand intérêt pédagogique. L'originalité de cet ouvrage réside dans l'esprit et dans la forme de la présentation. Son contenu est d'une grande qualité ; il est le fruit de trente années de réflexion et d'expérience pédagogique dans le domaine toujours d'actualité, relatif aux *outils pour la modélisation et la résolution des problèmes*. Ce sont des mathématiques discrètes pour l'ingénieur, l'informaticien et le chercheur.

Nous étions nombreux à attendre la parution de cet ouvrage et sommes reconnaissants à l'auteur, enseignant au Département d'Informatique de l'INSA, de l'avoir écrit. Nous le recommandons fortement aux élèves ingénieurs du Département et aux anciens élèves.

Professeur Jean-Marie PINON
Directeur du Département d'Informatique
de l'INSA de LYON

PREMIÈRE PARTIE

FONDEMENTS

CHAPITRE 0

MÉMENTO DE LOGIQUE

Ce chapitre peut être sauté en première lecture : il ne constitue qu'un bref aide-mémoire et n'a pour mission que de lever certains doutes que pourrait avoir le lecteur en théorie des ensembles et dans diverses démonstrations.

0.1 LOGIQUE DES PROPOSITIONS

On définit les propositions comme des énoncés, phrases décrivant une situation, une propriété, une relation, un jugement, auxquelles on serait susceptible d'associer dans des circonstances précises une valeur vrai/faux.

La logique se place essentiellement sous le signe de la *non-contradiction*. Pour bien contraster le vrai et le faux et éviter les paradoxes, on s'interdit les phrases à autoréférence telles que « cette phrase contient cinq mots » ou « quand je ne suis pas lue, je suis écrite en latin ».

Sous cette réserve, on suppose les propositions élémentaires traitées en bloc, sans attention aux sujets, compléments… comme pour l'énoncé « il fait beau ». C'est pourquoi ce niveau peut être mis en rapport direct avec l'algèbre de Boole.

0.1.1 Formalisation des propositions

DÉFINITION 0.1 – Les ***énoncés élémentaires*** seront représentés par des variables, dites ***variables propositionnelles***. Les ***énoncés composés*** seront formés d'un ou plusieurs énoncés (élémentaires ou composés) combinés par des ***connecteurs***:

L'énoncé composé	*est une*	*qui dénote un énoncé*
$A \wedge B \wedge C$	conjonction	vrai si et seulement si A et B et C sont tous vrais
$X \vee Y \vee Z$	adjonction[1]	vrai dès que l'un des X ou Y ou Z est vrai
$\neg D$	négation	vrai si D est faux, et réciproquement
$A \rightarrow D$	implication (matérielle)	vrai sauf si A est vrai et D faux
$A \leftrightarrow D$	double implication	vrai si A et D sont tous deux vrais, ou tous deux faux

Des combinaisons plus complexes pourront être formées par parenthèsage.

Ainsi : $((A \rightarrow D) \rightarrow (D \rightarrow B)) \rightarrow (A \rightarrow B)$

se lira si A alors D *implique que* si D alors B, *alors* A implique B.

---

[1] Le terme usuel est *disjonction*, mais il n'est pas approprié, et j'en déconseille l'usage.

0.1.2 Substitutions

DÉFINITION 0.2 – Soit E un énoncé contenant une variable propositionnelle p. L'énoncé obtenu en remplaçant dans E chaque occurrence de p par un énoncé F sera dit ***substitué*** de E, et noté E(p | F). On pourra procéder à plusieurs ***substitutions*** d'un seul coup s'il n'y a pas d'ambiguïté.

EXEMPLE. $E \equiv p \rightarrow (q \rightarrow p)$ donne $E(p \mid a \rightarrow b) \equiv (a \rightarrow b) \rightarrow (q \rightarrow (a \rightarrow b))$.

0.1.3 Règles d'inférences

DÉFINITION 0.3 – Toute logique possède des *mécanismes générateurs*, qui doivent permettre d'ajouter un énoncé à un contexte. Ces *figures de raisonnement licites* sont dites ***règles d'inférence.***

REMARQUES

- Les règles d'inférence sont notées $X, Y, Z \Rightarrow T$, pour signifier que l'énoncé de forme T, ou ***conséquent***, est valablement déduit d'énoncés de forme X, Y, Z, les ***antécédents.***
- Pour éviter les confusions, on réserve en général les majuscules aux (métavariables des) règles d'inférence, et les minuscules aux (variables des) énoncés spécifiques du problème.

0.1.4 Démonstrations

DÉFINITION 0.4 – Soit $E_1, E_2, \ldots, E_n$ une suite d'énoncés.

On dira qu'il s'agit d'une ***démonstration*** de ***germe*** $E_1, \ldots E_k$ si :

- $E_1, \ldots, E_k$ sont des énoncés au sens précédent,
- chaque énoncé E_i de la sous-suite d'énoncés $E_{k+1}, \ldots, E_n$ est dérivé de la sous-suite $E_1, \ldots, E_{i-1}$ au sens suivant :
  - il existe une *règle d'inférence* notée $X, Y, Z \Rightarrow T$,
  - il existe une *substitution* d'énoncés s telle que les substitués s(X), s(Y), s(Z) soient précisément des énoncés de $E_1, \ldots, E_{i-1}$;
  - et enfin, $E_i = s(T)$.

Alors, chacun des $E_{k+1}, \ldots, E_n$ est vrai si chacun des $E_1, \ldots, E_k$ est vrai.

0.1.5 Systèmes et démonstrations

Soit un certain système à étudier. Nous connaissons au moins certaines de ses *lois*, causales ou empiriques, et au moins partiellement son *état*.

Avec cela, on peut constituer un *germe* :

- en associant des énoncés élémentaires aux caractéristiques de l'état, une *situation* du système sera définie par une suite de ces énoncés ou de leurs négations ;
- les *lois* connues du système (causales ou factuelles) mèneront en général à des implications.

Dans ces conditions, une démonstration au sens ci-dessus fournit des énoncés vrais, applicables au système *dès lors* que le système vérifie les lois et l'état définis par le germe.

Si le germe se limite à des lois du système, les démonstrations (ne) produisent (que) des lois secondaires. Les résultats seront d'autant plus concrets que le germe comprendra une définition plus fine de l'état du système.

0.1.6 Principales règles d'inférence

Règles d'inférences positives, sans négation

Ces règles sont essentiellement celles des tableaux 0.1 et 0.2.

Tableau 0.1 Règles d'inférence positive.

Nom	Notation	Signification
recopie	$X \Rightarrow X$	
détachement	$X, X \rightarrow Z \Rightarrow Z$	si on a X, et «X implique Z», alors on a Z
conjonction	$X, Y \Rightarrow X \wedge Y$	si X, et Y, (isolément) alors « X et Y» (ensemble)
ventilation	$X \wedge Y \Rightarrow X, Y$	si «X et Y»(ensemble) alors X, et aussi Y (isolément)
conj + vent	$X \wedge Y \Leftrightarrow X, Y$	«X et Y» est vrai (ensemble) si et seulement si X et aussi Y sont vrais isolément
équivalence	$(X \leftrightarrow Y) \Leftrightarrow (X \rightarrow Y) \wedge (Y \rightarrow X)$	«X équivaut à Y» si, et seulement si «X implique Y» et «Y implique X»
adjonction (introduction)	$X \Rightarrow X \vee Y$	si X est vrai, «X ou Y» l'est.
dilemme	$(X \vee Y), (X \rightarrow R), (Y \rightarrow R) \Rightarrow R$	si «X ou Y», et «X implique R», et «Y implique R», alors R (sans avoir à déterminer lequel, de X, ou Y, est vrai)

Tableau 0.2 Règles d'articulation des démonstrations.

Nom	Notation	Signification
composition	$(X \Rightarrow Y, Y \Rightarrow Z) \Rightarrow (X \Rightarrow Z)$	si X entraîne Y, et si Y entraîne Z alors X entraîne Z
déduction	$(X \Rightarrow Y) \Rightarrow (X \rightarrow Y)$	Si X entraîne Y (par quelque raisonnement valide), alors le fait « X implique Y » est vrai

La règle de déduction engendre une implication comme *trace utile* d'une démonstration auxiliaire imbriquée localement dans la démonstration principale, ou faite séparément comme dans certains lemmes ou théorèmes techniques.

Règles d'inférence avec la négation

Le rôle de l'adjonction se précise. S'introduisent également des règles sur le thème *si la présence de A entraîne toujours celle de B, l'absence de B entraîne celle de A*.

On détecte les contradictions, notées $\perp$, et toute cause de contradiction est réfutée (tab. 0.3).

Tableau 0.3 Règles exploitant la négation.

Nom	Notation	Signification
adjonction / élimination	$(X \vee Y) \Rightarrow (\neg X \rightarrow Y), (\neg Y \rightarrow X)$	si X ou Y, alors si l'un est faux l'autre est vrai
contraposition	$(X \rightarrow Z) \Rightarrow (\neg Z \rightarrow \neg X)$	si un effet Z a une cause X, *alors* l'absence de l'effet implique l'absence de la cause
réfutation	$(X \rightarrow Z), \neg Z \Rightarrow \neg X$	si un effet a une cause, et si l'effet est absent, *alors* la cause est absente par contraposition + détachement
contradiction		
/ détection	$Z, \neg Z \Rightarrow \perp$	$\perp$ dénote la contradiction
/ élimination	$(X \rightarrow \perp) \Rightarrow \neg X$	ce qui mène à une contradiction est faux

Ces règles suffiront à établir au paragraphe 0.1.8 le principe de non-contradiction (ex. 6).

Règles aristotéliciennes

Avec l'involution, *la négation se durcit* ; la possibilité d'effacer les doubles négations donne des résultats plus forts car plus simples ; la contraposition est symétrisée, la réfutation engendre la *reductio ad absurdum* .

Ces règles, anciennes et efficaces, sont classiques en mathématiques, où la droite coupe ou non le plan. Elles ont été isolées car elles sont inadéquates pour modéliser les systèmes

- où les attributs ne sont pas bipolaires (X n'est ni pauvre ni riche...) ;
- où la double négation est une forme faible de l'affirmation (ce n'est pas inintéressant < c'est intéressant).

Tableau 0.4 Règles aristotéliciennes.

Nom	Notation	Signification
involution	$\neg\neg X \Rightarrow X$	double négation vaut affirmation
contraposition	$(X \rightarrow Z) \Leftrightarrow (\neg Z \rightarrow \neg X)$	symétrisation par effacement des doubles négations
absurdité	$\neg X \rightarrow \perp \Rightarrow X$	si nier une hypothèse entraîne une contradiction, l'hypothèse est établie (par réfutation + involution)

0.1.7 Inférences fondamentales et secondaires

Lorsqu'on rencontre plusieurs fois une même suite d'inférences ayant des effets globaux simples et caractérisés, on peut considérer ce stéréotype comme constituant une nouvelle règle d'inférence, ou ***règle d'inférence secondaire***, qui peut se révéler plus *commode* que les règles fondamentales.

Un raisonnement est valide s'il emploie des règles d'inférence valides, fondamentales ou secondaires. En pratique, l'homme a tendance à multiplier les stéréotypes pour aller plus vite. *Cependant, certains raisonnements sont faux (ou plutôt invalides) du fait de stéréotypes invalides.*

En cas de doute, *expliciter un raisonnement facilite sa vérification.*

0.1.8 Exemples

Nous numérotons les énoncés dans un style niveau.rang qui permet d'imbriquer des démonstrations. A chaque niveau, un trait horizontal séparera le germe (ou l'hypothèse) de sa dérivation. Une barre verticale commune marquera les énoncés d'une même démonstration locale.

Sans négation

EXEMPLE 1. Soit un système formellement régi par :

0.1 :	$h \to (r \vee p)$	/	énoncés initiaux
0.2 :	$r \to m$	/	
0.3 :	$p \to m$	/	...

Qu'en tirer ? h semble être la clé du système.

Posons :	1.0 : h	/	hypothèse pour une sous-démonstration
on en déduit	1.1 : $r \vee p$	/	détachement 0.1, 1.0 (X | h, Y | $r \vee p$)
puis	1.2 : m	/	dilemme 1.1, 0.2, 0.3 (X| h, Y| $r \vee p$, R| m)
d'où la propriété	*0.4 : $h \to m$*	/	déduction 1.0 – 1.2 (X| h, Y| m)

EXEMPLE 2. Montrer que l'on a toujours $(X \to (X \vee Y))$.

Partant d'un germe vide, nous tentons une sous-démonstration sur une proposition quelconque :

1.0	x	/	hypothèse niveau 1
1.1	$x \vee y$	/	adjonction à 1.0 (X|x, Y|y)
0.1	$x \to (x \vee y)$	/	déduction 1.0, 1.1 (X|x, Y|y)

Cette démonstration est correcte quelles que soient les propositions substituées à x et y, elle peut donc être vue comme tout à fait générale, et on retient son abstraction :

$$X \to (X \vee Y)$$

Ce résultat a la forme d'un énoncé, mais n'utilise que des métavariables : c'est un méta-énoncé qui vient d'être établi. Il constitue un ***métathéorème*** ou ***modèle de tautologie,*** fournissant des énoncés vrais pour toute substitution d'énoncé à X et/ou Y.

Un modèle de tautologie est assimilable à une règle d'inférence sans antécédent : tout substitué est une ***tautologie*** ou *énoncé universellement vrai*, i.e. indépendamment de la vérité de ses variables ; ces tautologies peuvent toujours être utilisées comme étape d'une démonstration.

Une tautologie est une cheville ou un *lieu commun* de la logique : n'apportant rien en elle-même, elle permet de relancer une démonstration, par combinaison avec des énoncés spécifiques au problème, et/ou en facilitant l'application de règles d'inférence plus constructives.

Ainsi peut-on, si nécessaire, insérer dans une démonstration un énoncé tel que

$$1.17 : ((a\to b)\to c) \to (((a\to b)\to c)\vee(p\wedge a))$$

comme *tautologie* obtenue par substitution de $((a\to b)\to c)$ à X, et de $(p\wedge a)$ à Y, dans le *modèle de tautologie* $X\to(X\vee Y)$.

En remplaçant l'adjonction en 1.1 par d'autres figures, on obtiendra d'autres modèles de tautologies, tels que :

$(X \to X)$	par simple recopie,
$(X \to (X \wedge X))$	par conjonction,
$(X \wedge Y) \to X$	par ventilation,
$(X \vee X) \to X$	par dilemme...

EXEMPLE 3. Montrer que $X\to(X\to Y) \Rightarrow X\to Y$.

0.1	$(x\to(x\to y))$	hypothèse initiale
1.0	x	hypothèse niveau 1
1.1	$(x\to y)$	détachement 0.1, 1.0 $(X \mid x, Y \mid (x \to y))$
1.2	y	détachement 1.1, 1.0 $(X \mid x, Y \mid y)$
0.2	$(x\to y)$	déduction 1.0, 1.2 $(X \mid x, Y \mid y)$

Cette démonstration n'a rien de circonstanciel. Pouvant supporter toute substitution de proposition aux variables propositionnelles x et y, elle peut être considérée comme faisable en toute généralité. L'infinité des démonstrations possibles se résume par l'abstraction obtenue en substituant des métavariables aux variables :

$$(X\to(X\to Y)) \Rightarrow (X\to Y)$$

Cette règle introduit un procédé de *simplification d'énoncé*, et aussi le *modèle de tautologie*

$$((X\to(X\to Y))\to(X\to Y))$$

EXEMPLE 4. SYLLOGISME. Montrer que $(A\to B), (B\to C) \Rightarrow (A\to C)$.

La démonstration ci-après pouvant supporter toute substitution de proposition aux variables a, b, c, elle pourra être considérée comme faisable en toute généralité.

0.0	$(a \to b)$	germe
0.1	$(b \to c)$	germe
1.0	$\vert\ a$	hypothèse
1.1	$\vert\ b$	détachement 0.0, 1.0
1.2	$\vert\ c$	détachement 0.1, 1.1
0.3	$(a \to c)$	déduction 1.0, 1.2

On pourra admettre comme règle d'inférence son abstraction :

$$(A \to B), (B \to C) \Rightarrow (A \to C)$$

Le procédé s'étend aux ***sorites*** ou *syllogismes en chaîne*

$$(A \to B), (B \to C), (C \to D), (D \to E), (E \to F) \Rightarrow (A \to F)$$

(cf. Lewis CAROLL et § 0.3.10).

Des démonstrations semblables engendrent des tautologies apparentées :

$$((A \to B) \wedge (B \to C)) \to (A \to C)$$
$$((A \to B) \to ((B \to C) \to (A \to C))) \ldots$$

Avec négation, sans involution

EXEMPLE 5. Montrer que $(A \to \neg A) \Rightarrow \neg A$.

0.1	$(a \to \neg a)$	germe
1.0	$\vert\ a$	hypothèse
1.1	$\vert\ \neg a$	détachement 0.1, 1.0 $(X\vert a, Y\vert \neg a)$
1.2	$\vert\ \bot$	détection contradiction 1.0, 1.1
0.2	$(a \to \bot)$	déduction 1.0 à 1.2
0.3	$\neg a$	élimination contradiction

Cette démonstration, supportant toute substitution sur a, peut être considérée comme faisable en toute généralité, et on peut admettre son abstraction :

$$(A \to \neg A) \Rightarrow \neg A$$

Plus directement, on pourrait écrire la méta-déduction :

0.1	$(A \to \neg A)$	germe
1.0	$\vert\ A$	hypothèse
1.1	$\vert\ \neg A$	détachement 0.1, 1.0
1.2	$\vert\ \bot$	détection contradiction 1.0, 1.1
0.2	$(A \to \bot)$	déduction
0.3	$\neg A$	élimination contradiction

Résumée par : $(A \to \neg A) \Rightarrow \neg A$
qui mène au schéma de tautologie : $((A \to \neg A) \to \neg A)$

EXEMPLE 6. PRINCIPE DE NON-CONTRADICTION (LUKASIEWICZ).

Le principe « *une proposition et sa contraire ne sont jamais simultanément vraies* » s'établit en supposant une telle conjonction, qui entraîne une contradiction et se trouve ainsi réfutée.

1.0	$(A \wedge \neg A)$	hypothèse
1.1	A	ventilation 1.0
1.2	$\neg A$	ventilation 1.1
1.3	$\perp$	contradiction 1.1, 1.2
0.0	$\mathbf{\neg (A \wedge \neg A)}$	réfutation 1.0, 1.3

Ce *principe fondamental* découle de la seule réfutation.

EXEMPLE 7. PREMIÈRE RÈGLE DE DE MORGAN $\neg(X \vee Y) \rightarrow (\neg X \wedge \neg Y)$.

1.0	X	hypothèse
1.1	$X \vee Y$	adjonction 1.0
0.0	$X \rightarrow (X \vee Y)$	déduction 1.0 1.1
1.0	Y	hypothèse
1.1	$X \vee Y$	adjonction 1.0
0.1	$Y \rightarrow (X \vee Y)$	déduction 1.0 1.1
1.0	$\neg(X \vee Y)$	hypothèse
1.1	$\neg X$	contraposition 0.0, 1.0
1.2	$\neg Y$	contraposition 0.1, 1.0
1.3	$(\neg X) \wedge (\neg Y)$	conjonction 1.2 1.3
0.2	$\mathbf{\neg(X \vee Y) \rightarrow ((\neg X) \wedge (\neg Y))}$	déduction 1.0 à 1.3

Avec négation involutive

EXEMPLE 8. PRINCIPE DU TIERS EXCLUS *« tertium non datur »*.

Le principe « *d'une proposition et sa contraire, au moins l'une est vraie* », noté $A \vee \neg A$, s'établit par l'absurde, en montrant que $\neg(A \vee \neg A)$ entraîne une contradiction sur $\neg A$.

1.0	A	hypothèse
1.1	$A \vee \neg A$	adjonction 1.0
0.0	$A \rightarrow (A \vee \neg A)$	déduction 1.0 1.1
1.0	$\neg A$	hypothèse
1.1	$A \vee \neg A$	adjonction 1.0
0.1	$\neg A \rightarrow (A \vee \neg A)$	déduction 1.0 1.1

1.0	$\neg(A \vee \neg A)$	hypothèse
1.1	$\neg A$	contraposition 0.0, 1.0
1.2	$\neg\neg A$	contraposition 0.1, 1.0
1.3	$\perp$	contradiction sur $\neg A$, 1.1 1.2
0.2	$\neg\neg(A \vee \neg A)$	réfutation 1.0, 1.3
0.3	$\mathbf{(A \vee \neg A)}$	involution 0.2, ou absurdité 1.0 1.3

EXEMPLE 9. SECONDE RÈGLE DE DE MORGAN : $\neg(X \wedge Y) \rightarrow (\neg X \vee \neg Y)$.

1.0	$\neg X$	hypothèse
1.1	$\neg X \vee \neg Y$	adjonction 1.0
0.0	$\neg X \rightarrow (\neg X \vee \neg Y)$	déduction 1.0 1.1
1.0	$\neg Y$	hypothèse
1.1	$\neg X \vee \neg Y$	adjonction 1.0
0.1	$\neg X \rightarrow (\neg X \vee \neg Y)$	déduction 1.0 1.1
1.0	$\neg(\neg X \vee \neg Y)$	hypothèse
1.1	$\neg\neg X$	contraposition 0.0, 1.0
1.2	$\neg\neg Y$	contraposition 0.1, 1.0
1.3	X	involution 1.1
1.4	Y	involution 1.2
1.5	$X \wedge Y$	conjonction 1.3 1.4
0.2	$\neg(\neg X \vee \neg Y) \rightarrow (X \wedge Y)$	déduction 1.0 1.5
0.3	$\neg(X \wedge Y) \rightarrow \neg\neg(\neg X \vee \neg Y)$	contraposition 0.2
0.4	$\mathbf{\neg(X \wedge Y) \rightarrow (\neg X \vee \neg Y)}$	involution 0.3

0.2 LOGIQUE DES PRÉDICATS

C'est la logique la plus utilisée, ni trop simpliste, ni trop complexe.

0.2.1 Prédicats

On s'intéresse toujours aux phrases auxquelles on devrait pouvoir associer dans des circonstances précises une valeur vrai/faux.

Les ***énoncés élémentaires*** ou ***prédicats*** portent maintenant sur la vérité/le jugement d'une ***relation*** entre des ***objets*** ou ***termes***, correspondants aux sujets, compléments... de la phrase.

Ainsi, « Pierre est au jardin » pourra être ici noté *estEn(Pierre, jardin)* où *estEn* désignera une *relation* entre un *objet* (ici *Pierre*) et son *emplacement* (ici *jardin*).

0.2.2 Formalisation des termes

DÉFINITION 0.5 – Un ***terme*** est un *objet* ou *individu* dénoté :

- soit par une ***constante*** ; ex : Pierre, jardin,
- soit par une ***variable*** ; ex : x, y...
- soit par une ***notation fonctionnelle*** $f(t_1, t_2, \ldots, t_k)$, où f est un symbole fonctionnel ou nom de fonction, et où les $t_1, t_2, \ldots, t_k$ sont des termes ;
  ex : père(Pierre), premier(enfants(Pierre, Marie)).

On appelle ***terme clos*** un terme n'utilisant pas de variables, et qui dénote ainsi indirectement des constantes.

0.2.3 Formalisation des énoncés

Enoncés élémentaires

DÉFINITION 0.6 – En logique des prédicats, un ***énoncé élémentaire*** ou ***prédicat atomique*** est une écriture fonctionnelle $p(t_1, t_2, \ldots, t_k)$

- où p est un symbole prédicatif ou nom de prédicat,
- et dont les arguments $t_1, t_2, \ldots, t_k$ sont des termes.

Un tel énoncé

- *dénote une relation* précisée par le symbole prédicatif, et portant sur les termes mentionnés ;
- a pour valeur non un terme mais une *valeur de vérité* associée à cette relation, satisfaite ou non par ce jeu de termes.

EXEMPLES

estEn(père(Pierre), jardin) pour « le père de Pierre est au jardin » (vrai/faux),
plus(X, 0, X) pour « X + 0 = X » (vrai/faux).

Enoncés composés

DÉFINITION 0.7 – Plus généralement on appellera ***énoncé*** :

- tout énoncé élémentaire,
- si E, E_1, E_2 sont des énoncés, l'une des notations du tableau 0.5.

Tableau 0.5 Énoncés composés.

Notation	Signification	
$(E_1 \wedge E_2)$	E_1 et E_2	
$(E_1 \vee E_2)$	E_1 ou E_2	
$(E_1 \rightarrow E_2)$	E_1 implique E_2,	
$(E_1 \leftrightarrow E_2)$	E_1 équivaut à E_2 ; E_1 si et seulement si E_2, en abrégé « E_1 ssi E_2 »	
$\neg E$	non E	
$\forall x\ E(x)$	pour tout x, E(x)	/ où x est une variable muette
$\exists x\ E(x)$	il existe au moins un x tel que E(x),	/ où x est une variable muette

Les symboles « $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$ » sont des *connecteurs,* les nouveaux symboles « $\forall$, $\exists$ » sont dits ***quantificateurs.***

Les deux dernières formes, dites *quantifiées*, généralisent l'énoncé interne.

Les énoncés	*s'interprètent*
$\forall x$ plus(x,0,x)	pour tout x, x + 0 = x
$\forall x\ \exists y$ plus(x,y,0)	pour tout x, il existe au moins un y tel que x + y = 0
$\forall x$ (homme(x)$\rightarrow$mortel(x))	pour tout x, si x est un homme alors x est mortel, ou : tout homme est mortel.

Lexiques

Pour éviter toute confusion, on utilise des *lexiques disjoints* pour les *variables,* les *fonctions*, les *prédicats*.

Arités

DÉFINITION 0.8 – On appelle ***arité*** d'une fonction ou d'un prédicat, le nombre de termes qu'il faut associer à son nom.

Le lexique des *constantes symboliques* s'identifie au lexique des fonctions d'arité 0.

On appelle souvent *propositions* les prédicats d'arité 0, et ***propriétés*** les prédicats d'arité 1.

0.2.4 Systèmes et démonstrations

Comme précédemment on modélise un système en exprimant sous forme logique ses *lois* et son *état.*

Ceci constituera le germe de démonstrations exploitant les règles d'inférence.

Si le germe est réduit aux *lois* du système, la démonstration établit des *lois secondaires* du système. Si le germe contient des *énoncés sur l'état* du système, on obtient des *résultats factuels.*

0.2.5 Règles d'inférence

Comme dans la section 0.1, elles sont notées *antécédents* $\Rightarrow$ *conséquent*, et signifient : *si le contexte satisfait les antécédents à une substitution* s *près, alors on peut,* dans la logique considérée, *ajouter au contexte le conséquent soumis à la même substitution* s, où le contexte désigne les énoncés du germe, augmenté des conséquents déjà établis.

Les *nouveautés* sont relatives aux *quantificateurs*, et à la nécessité pour les substitutions d'*accorder* entre eux les *énoncés* d'une part, les *termes* de l'autre.

Pour éviter une définition trop complexe des ces règles, commençons par celles qui permettent de faire apparaître ou d'éliminer

- les connecteurs,
- les quantificateurs (en rapport avec les termes).

Nous pouvons utiliser le système du tableau 0.6, où le symbole λ dénote une *contradiction*, complété par les tableaux 0.7 et 0.8.

Tableau 0.6 Règles d'inférence sur les connecteurs.

Nom	Introduction	Elimination
implication	$(A \Rightarrow B) \Rightarrow (A \rightarrow B)$	$A, (A \rightarrow B) \Rightarrow B$
conjonction	$A, B \Rightarrow (A \wedge B)$	$(A \wedge B) \Rightarrow A, B$
équivalence	$(A \rightarrow B) \wedge (B \rightarrow A) \Rightarrow (A \leftrightarrow B)$	$(A \leftrightarrow B) \Rightarrow (A \rightarrow B) \wedge (B \rightarrow A)$
contradiction	$B, \neg B \Rightarrow \bot$	réfutation $(A \Rightarrow \bot) \Rightarrow \neg A$ absurdité $(\neg A \Rightarrow \bot) \Rightarrow A$
involution	$A \Rightarrow \neg\neg A$	$\neg\neg A \Rightarrow A$
adjonction	$A \Rightarrow (A \vee B), (B \vee A)$	par dilemme $(A \vee B), (A \rightarrow C), (B \rightarrow C) \Rightarrow C$ par implications $(A \vee B) \Rightarrow (\neg A \rightarrow B), (\neg B \rightarrow A)$

Tableau 0.7 Règles sur les quantificateurs.

Quantification	Généralisation	Spécification
universelle	$A(t) \Rightarrow \forall x\, A(x)$, où t est une variable libre	$\forall x\, A(x) \Rightarrow A(t)$, où t est un terme quelconque
existentielle	$A(a) \Rightarrow \exists x\, A(x)$, où a est un terme clos	$\exists x\, A(x) \Rightarrow A(\alpha)$, où α est une nouvelle constante, ajoutée au lexique

Tableau 0.8 Règles globales.

Nom	Enoncés
dualité	$\forall x\, E(x) \Leftrightarrow \neg\exists x\, \neg E(x)$ $\neg\forall x\, E(x) \Leftrightarrow \exists x\, \neg E(x)$ $\exists x\, E(x) \Leftrightarrow \neg\forall x\, \neg E(x)$
récurrence	$E(0), \forall n\, (E(n) \rightarrow E(n+1)) \Rightarrow \forall n\, E(n)$

0.2.6 Exemples

EXEMPLE 10. Prouver que si un entier a n'est pas divisible par 3, alors a^2-1 est divisible par 3.

Notons mod(a, p, r) le prédicat « a divisé par p a pour reste r ».

0.1	$\forall a \forall b \forall p\ \mathrm{mod}(a, p, 0) \rightarrow \mathrm{mod}(a \cdot b, p, 0)$	def mod
0.2	$\forall a\ \mathrm{mod}(a, 3, 0) \vee \mathrm{mod}(a-1, 3, 0) \vee \mathrm{mod}(a+1, 3, 0)$	spec mod 3
0.3	$\forall a \forall b \forall p\ \mathrm{mod}((a-1), p, 0) \rightarrow \mathrm{mod}((a-1) \cdot b, p, 0)$	0.1 $(a \mid a-1)$
0.4	$\forall a \forall p\ \mathrm{mod}((a-1), p, 0) \rightarrow \mathrm{mod}((a-1) \cdot (a+1), p, 0)$	0.3 $(b \mid a+1)$
0.5	$\forall a \forall p\ \mathrm{mod}((a-1), p, 0) \rightarrow \mathrm{mod}(a^2-1, p, 0)$	propriété de $\cdot$
0.6	$\forall a \forall b \forall p\ \mathrm{mod}((a+1, p, 0) \rightarrow \mathrm{mod}((a+1) \cdot b, p, 0)$	0.1 $(a \mid a+1)$
0.7	$\forall a \forall p\ \mathrm{mod}((a+1), p, 0) \rightarrow \mathrm{mod}((a+1) \cdot (a-1), p, 0)$	0.6 $(b \mid (a-1))$
0.8	$\forall a \forall p\ \mathrm{mod}((a+1), p, 0) \rightarrow \mathrm{mod}(a^2-1, p, 0)$	propriété de $\cdot$
0.9	$\forall a\ \neg\mathrm{mod}(a, 3, 0) \rightarrow (\mathrm{mod}(a-1, 3, 0) \vee \mathrm{mod}(a+1, 3, 0))$	élim. adj.0.2

1.0	$\neg mod(a,3,0)$	hypothèse
1.1	$\neg mod(a, 3, 0) \rightarrow (mod\ a-1, 3, 0) \vee mod(a+1, 3, 0))$	spec 0.9
1.2	$(mod(a-1, 3, 0) \vee mod(a+1, 3, 0))$	détachement 1.0, 1,1
1.3	$mod((a-1), 3, 0) \rightarrow mod(a^2-1, 3, 0)$	spec 0.5
1.4	$mod((a+1), 3, 0) \rightarrow mod(a^2-1, 3, 0)$	spec 0.8
1.5	$mod(a^2-1, 3, 0)$	dilemme 1.2-1.3-1.4
0.10	$\neg\ mod(a,3,0) \rightarrow mod\ (a^2-1, 3, 0)$	déduction 1.0 -1.5
0.11	$\forall a\ (\neg mod(a,3,0) \rightarrow mod(a^2-1, 3, 0))$	généralisation 0.10

EXEMPLE 11. Montrer qu'une relation R *intransitive* est *irréflexive.*

Argument : si un élément a est en relation avec lui-même par R, il l'est autant de fois que l'on veut ; or s'il l'est deux fois, l'intransitivité interdit une troisième, d'où contradiction.

0.1	**$\forall a \forall b \forall c\ (R(a, b) \wedge R(b, c)) \rightarrow \neg R(a, c)$**	**def. intransitivité**
0.2	$\forall a \forall b\ (R(a, b) \wedge R(b, b)) \rightarrow \neg R(a, b)$	0.1 spec univ (c| b)
0.3	$\forall a\ (R(a, a) \wedge R(a, a)) \rightarrow \neg R(a, a)$	0.2 spec univ (b| a)
1.0	$R(a, a)$	hypothèse locale
1.1	$R(a, a) \wedge R(a, a)$	conjonction 1.0, 1.0
1.2	$(R(a, a) \wedge R(a, a)) \rightarrow \neg R(a, a)$	spec.univ. 0.3
1.3	$\neg R(a, a)$	détachement 1.1, 1.2
1.4	$\perp$	contradiction 1.0, 1.3
0.4	$\neg R(a, a)$	réfutation 1.0,1.4
0.5	$\forall a\ \neg R(a, a)$	généralisation univ. 0.4 ≡ **définition irréflexivité attendue**

0.3 EXERCICES

0.3.0

Montrer que $(A \wedge B) \rightarrow (A \vee B)$.

0.3.1

Montrer que $A \rightarrow B,\ B \rightarrow C,\ C \rightarrow \neg A \Rightarrow \neg A$.
A est-il condition nécessaire ou suffisante de B ?

0.3.2

Montrer que $A \rightarrow P,\ B \rightarrow P \Rightarrow (A \vee B) \rightarrow P$ et que $P \rightarrow X,\ P \rightarrow Y \Rightarrow P \rightarrow (X \wedge Y)$.
Quelles sont les *conditions suffisantes* de P ? Ses *conditions nécessaires* ?

Montrer que

$$A \to P, B \to P, P \to X, P \to Y, (X \wedge Y) \to (A \vee B) \Rightarrow$$
$$(X \wedge Y) \leftrightarrow P, P \leftrightarrow (A \vee B), (X \wedge Y) \leftrightarrow (A \vee B).$$

0.3.3

On suppose que $A \to B$ signifie que A est *une* condition suffisante de B. Que signifie la contraposition de cette implication ? Peut-on dire que B est *condition nécessaire* de A ?

0.3.4

La règle *« ex falso quodlibet sequitur »,* qu'on peut transcrire $\perp \to A$, caractérise les systèmes tolérant la contradiction.

Montrer que, si la contradiction est admise, on obtient

- par réfutation, $\perp \Rightarrow \neg A$ (toute proposition est fausse),
- par absurdité, $\perp \Rightarrow A$ (toute proposition est vraie),
- et que le fait qu'une proposition soit dérivable n'est plus informatif.

En déduire les comportements à adopter

- si un germe et une hypothèse locale entraînent une contradiction,
- si un germe entraîne une contradiction sans hypothèse locale particulière,
- si une contradiction est dérivable *ex nihilo*.

0.3.5

Montrer que $A, A \leftrightarrow B \Rightarrow B$.

0.3.6

Montrer que $A \leftrightarrow B, B \to C, C \leftrightarrow D, D \to \neg A \Rightarrow \neg A, \neg B$.

0.3.7

Trois personnes de trois nationalités différentes habitent les trois premières maisons d'une rue. Chaque maison a une couleur différente, et chaque personne un métier différent. On constate que :

Le Français habite la maison rouge.
L'Allemand est musicien.
L'Anglais habite la seconde maison.
La maison rouge est à côté de la verte.
L'écrivain habite la première maison.

- Quelle est la nationalité de l'écrivain ?
- Qui habite la maison jaune ?
- Où habite le peintre ?

Dresser maintenant un bilan général de la situation.

0.3.8

On considère la suite de Fibonacci définie par :

$$f(0) = 0 \text{ ; } f(1) = 1 \text{ ; au-delà, } f(i) = f(i-1) + f(i-2)$$

Montrer par récurrence que l'on a toujours

- $\forall I \in \mathbb{N}, f(i) \in \mathbb{N}$,
- $\forall I \in \mathbb{N}, f(i) \leq \exp(i/2)$,
- $\forall i \in \mathbb{N}, f(i) = a^i + b^i$, où a et b sont les deux solutions de $x^2 - x - 1 = 0$.

0.3.9

Partant de $P(0) \equiv 1 = 2 - 1$, montrer par récurrence le prédicat $P(n) \equiv 1 + 2 + 4 + \ldots + 2^{n-1} + 2^n = 2^{n+1} - 1$.

Jadis, aux Indes, l'inventeur du jeu d'échecs en fit une démonstration à un puissant radjah. Celui-ci demanda à l'inventeur quelle récompense il voulait pour avoir inventé un jeu aussi divin. L'inventeur demanda seulement 1 grain de blé pour la première case, 2 pour la seconde, 4 pour la troisième en doublant ainsi de suite jusqu'à la 64e. A raison de 0,1 gramme par grain, combien cela représente-t-il en tout de millions de tonnes de blé ?

0.3.10

Sachant que

a. *Les dentistes font les meilleurs rugbymen.*
b. *Les Ecossais portent des chaussettes rouges.*
c. *Ceux qui élèvent des taureaux ne portent rien de rouge.*
d. *Les meilleurs rugbymen sont Ecossais.*

Montrer que :

e. *Les Ecossais n'élèvent pas de taureaux.*
f. *Les meilleurs rugbymen portent des chaussettes rouges.*
g. *Les meilleurs rugbymen n'élèvent pas de taureaux.*
h. *Les dentistes sont Ecossais.*
i. *Les dentistes portent des chaussettes rouges.*
j. *Les dentistes n'élèvent pas de taureaux.*

0.4 PROBLÈMES

0.4.0

En étendant par récurrence l'exemple 10 du paragraphe 0.2.6, montrer que pour $n \geq 1$:

- si un entier a n'est pas divisible par 3, alors $a^{2n} - 1$ est divisible par 3 ;
- si un entier a est congru à 1 modulo p, alors $a^n - 1$ est divisible par p .

0.4.1

Montrer par l'absurde que $\sqrt{2}$ est irrationnel (Euclide).

0.4.2

Montrer par l'absurde que l'ensemble des nombres premiers n'est pas borné, i.e. qu'il n'y a pas de plus grand nombre premier (Euclide).

0.4.3

Montrer que toute proposition est logiquement encadrée par

- l'adjonction de ses conditions suffisantes, la plus forte de ses définitions par défaut,
- la conjonction de ses conditions nécessaires, la plus faible de ses définitions par excès.

0.4.4

Une fonction sera dite callimorphe si elle atteint son maximum sur la frontière F de son domaine D. On supposera démontré que si f est callimorphe, – f l'est aussi.

En déduire que si f est callimorphe, elle atteint son minimum à la frontière F de son domaine D.

Aussi, que si une fonction callimorphe est constante sur la frontière F de son domaine D, elle est constante dans tout le domaine D (Lejeune-Dirichlet).

0.4.5

Sur l'Ile aux Chevaux, les chevaux détiennent la parole, la sagesse, la vérité et la vertu, tandis que les humains y sont des bêtes de somme répugnantes et muettes. Comment Gulliver, l'étrange humain-qui-parle va-t-il justifier aux Principaux des chevaux son arrivée sur l'île ? Son naufrage conclut un voyage improvisé du fait de la trahison d'un ami. Et les chevaux sont ici de redoutables logiciens, qui ignorent cependant tout de l'erreur et du mensonge (d'après Jonathan Swift).

0.4.6

A Ivanbourg, le Camarade Premier Secrétaire, du fait de ..., est nécessairement le plus intelligent de tous les habitants. D'autre part, du fait de ..., le Camarade Premier Secrétaire est nécessairement le moins intelligent de tous les habitants.

En déduire que tous les habitants sont aussi intelligents les uns que les autres (d'après Zinoviev, *Les Hauteurs Béantes*, Ed. L'Age d'Homme).

0.4.7

Dans ce lointain pays, on s'accorde dans les milieux gouvernementaux à penser que:

- pour gagner les élections, il faut réduire le chômage ;
- pour réduire le chômage, il faut développer l'enseignement professionnel et réduire les impôts ;
- pour développer l'enseignement professionnel, il faut plus de fonds publics ;
- pour disposer de plus de fonds publics, il faut privatiser les firmes nationalisées, ou augmenter les impôts.

Formaliser ces règles en posant : g ≡ gagner les élections, r ≡ réduire le chômage...

Ce gouvernement ayant l'intention de gagner les élections, en déduire qu'il va privatiser les firmes nationalisées.

Revoir d'un œil critique les différents stades : énoncé phrasé des règles, formalisation des règles, démonstration. Examiner l'impact d'un changement dans l'une des 4 règles énoncées.

Réduire et augmenter les impôts sont-ils la parfaite négation l'un de l'autre?

0.5 PISTE DE RÉFLEXION

Comment discuter dans une langue où il est extrêmement impoli de dire non? Où il n'y a pas de négation ?

0.6 LECTURES

AHO A., ULLMAN J., *Concepts fondamentaux de l'informatique*, chap. XII et XIV, Dunod, Paris, 1993.

BENZAKEN C., *Systèmes formels*, Masson, Paris, 1991.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science*, vol. I, chap. IV.8, Coll. Texts and Monographs on Computer Science, Springer Verlag, NY & Berlin, 1991.

GRIZE J.B., *Logique moderne*, 3 fascicules, Mouton & Gauthier-Villars, 1973.

PABION J.F., *Logique mathématique,* Hermann, Paris, 1972.

REINHARDT R.& SOEDER H., *Atlas des mathématiques*, chap. I, La Pochothèque, Le Livre de Poche, Paris, 1997 ; traduction de *DTV-Atlas der Mathematik*, Deutscher Taschenbuch Verlag Gmbh, Munich, 1974.

VÉLU J., *Méthodes mathématiques pour l'informatique*, chap. 13, Dunod, Paris, 1994.

Collectif, *Petite encyclopédie des mathématiques*, chap. 15, Edition Pagoulatos, Paris, 1980 ; traduction de *Kleine Enzyklopädie der Mathematik*, VEB Bibliographisches Institut, Leipzig, 1975.

Curiosité :

CAROLL L. (DODGSON Ch. dit), *Logique sans peine,* Hermann, Paris, 1966 ; sélection de textes de logique des années 1890 ; normalisation des énoncés textuels, sorites, paradoxes...

CHAPITRE 1

ENSEMBLES & ÉLÉMENTS

Par la suite, un grand nombre de notations ou de concepts seront introduits par l'écriture :

Nouveau concept ou nouvelle notation ≡ développement

1.1 PRÉSENTATION

DÉFINITION 1.1 – Un ***ensemble*** est un regroupement en une entité d'objets distincts.

Cette définition, considérée comme première, ne sera pas autrement justifiée.

Les objets appartenant à un ensemble sont dits ***éléments*** ou ***membres*** *de l'ensemble.*

1.1.1 Extension, appartenance

DÉFINITION 1.2 – Un ensemble est dit ***ensemble défini en extension*** s'il est défini par la liste de ses éléments.

On notera :

ClubDesAmandiers = { Julie, Marcel, André, Sophie } ≡
« Julie, Marcel, André et Sophie forment le Club des Amandiers »

ClubDesAmandiers désigne l'*ensemble* des personnages.
Les membres du club Julie, Marcel, André, Sophie *appartiennent* chacun au club, et forment ensemble son *extension*.

$e_i \in E \equiv$ « e_i ***appartient à*** E », ainsi Sophie ∈ ClubDesAmandiers.
$z \notin E \equiv$ « z ***n'appartient pas à*** E », ainsi Zoé ∉ ClubDesAmandiers.

1.1.2 Cardinalité

DÉFINITION 1.3 – Pour un ensemble E défini en extension, on appelle ***cardinalité*** le nombre d'éléments de sa liste, notée |E|.

Ainsi, $E = \{e_1, e_2, e_3, e_4\}$ entraîne $|E| = 4$.

1.1.3 Identité

DÉFINITION 1.4 – Les deux ensembles A et B sont des ***ensembles identiques***, si tout élément de A est élément de B et, *réciproquement*, si tout élément de B est élément de A ; autrement dit, *si A et B ont même extension* (Leibniz). On pose :

$$A = B \equiv \forall x\ (x \in A \leftrightarrow x \in B)$$

et $A \neq B$ si les extensions diffèrent.

PROPOSITION 1.1 – $A = B \rightarrow |A| = |B|$.

1.1.4 Diagramme d'Euler

Pour raisonner sur de tels ensembles, on peut utiliser un ***diagramme d'Euler***, figure plane dans laquelle un ensemble E est représenté par une tache plus ou moins régulière, et, s'il y a lieu, les éléments par des points.

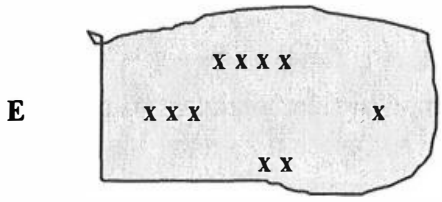


Fig. 1.0 Ensemble E et ses éléments.

1.1.5 Abstraction

L'énumération peut être lassante ou impossible.

DÉFINITION 1.5 – La notation E = { x | K(x) } définit E comme *ensemble des x tels que K(x).*

La notation peut être associée à un procédé constructif, K caractérisant les objets construits. Cette notation est formelle et non factuelle : c'est seulement si K n'est pas faux pour tout x que le procédé sera productif, et E effectif.

1.1.6 Compréhension

Il sera souvent plus commode ou intéressant de stipuler un ensemble E comme restriction qualitative d'un ensemble U plus vaste.

DÉFINITION 1.6 – On dira un ensemble E ***ensemble défini en compréhension*** si E est défini comme l'ensemble des x d'un ensemble de référence U tels que p(x), où p(x) est un prédicat défini dans U, et on notera $E = \{ x \in U \mid p(x) \}$.

L'appartenance à cet ensemble U d'objets peut être vue comme *condition nécessaire* d'appartenance à E, le *raffinement* de U par p menant à l'ensemble E visé.

Le *diagramme d'Euler* de la figure 1.2 représente E par une tache plus ou moins régulière *interne* à celle représentant l'ensemble U dont E est une restriction.

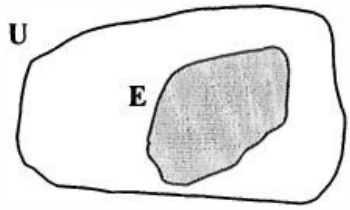


Fig. 1.1 Ensemble en compréhension.

PROPOSITION 1.2

$E = \{ x \in U \mid p(x) \} \Rightarrow |E| \leq |U|$

1.1.7 Ensemble vide

DÉFINITION 1.7 – L'***ensemble vide*** ou ensemble à 0 élément, est noté $\varnothing$.

PROPOSITION 1.3 – $\varnothing \equiv \{\}$ et $|\varnothing| = 0$.

En termes d'abstraction / compréhension, $\varnothing$ est l'*ensemble des objets réalisant un prédicat toujours faux.*

NOTE. Logique et théorie des ensembles classiques reposent sur un principe de non-contradiction. Aucun objet ne peut satisfaire une proposition contradictoire, ou si l'on préfère, seuls les 0 objets de l'ensemble vide le font.

1.1.8 Autres ensembles remarquables

Certains ensembles sont si remarquables qu'on leur a attribué un nom (en fait, une lettre) qui bénéficie d'une *typographie spéciale* pour éviter toute confusion avec des homonymes. Par la suite, nous noterons donc :

$\mathbb{B} = \{0, 1\}$, l'ensemble des ***booléens*** ;

$\mathbb{N}$, l'ensemble des ***entiers naturels*** ; il contient 0, et s'il contient n, il contient n + 1 ;

$\mathbb{Z}$, l'ensemble des ***entiers algébriques ou signés*** ; il contient $\mathbb{N}$, et s'il contient x et y, alors il contient x – y ;

$\mathbb{Q}$, l'ensemble des ***rationnels*** ; il contient $\mathbb{Z}$, et s'il contient p, et q non nul, alors il contient la forme irréductible de la fraction p/q.

$\mathbb{R}$, l'ensemble des ***réels,*** qui contient $\mathbb{Q}$ et les limites des suites convergentes à valeurs dans $\mathbb{Q}$;

$\mathbb{C}$, l'ensemble des ***complexes*** (x + iy), où x et y sont des réels.

1.2 INCLUSIONS

1.2.1 Inclusion large

DÉFINITION 1.8 – L'ensemble E est ***inclus au sens large*** dans l'ensemble F (resp. F ***contient*** E) *ssi* tout élément de E est élément de F. Ce qu'on note :

$$E \subseteq F \equiv F \supseteq E \equiv \forall x\ (x \in E) \rightarrow (x \in F)$$

Avec $E \subseteq F$, E est encore dit ***sous-ensemble*** (au sens large), ou ***partie*** de F.

Dans le cas contraire, on notera $E \not\subset F$.

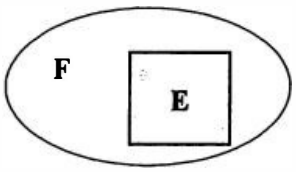


Fig. 1.2 Inclusion de E dans F.

PROPOSITIONS

1.4 – Pour tout ensemble E,

- $\varnothing \subseteq E$
- $E \subseteq E$ /réflexivité

PREUVE. Toute proposition fausse implique toute autre, au moins aussi vraie ! Donc $\forall x\ x \in \varnothing \rightarrow x \in E$. D'autre part, $E \subseteq E$ découle directement de la définition 1.8.

1.5 – Pour tous les ensembles E, F : $E \subseteq F \Rightarrow |E| \leq |F|$.

1.6 – Pour tous les ensembles E, F, G :

$E \subseteq F, F \subseteq G \Rightarrow E \subseteq G$, /transitivité
$E \subseteq F, F \subseteq E \Rightarrow E = F$ /antisymétrie

1.7 – $\subseteq$ est un ***ordre large*** *entre ensembles.*

NOTE. Réflexivité, transitivité, antisymétrie, ordre large signalés ici seront vus au chapitre 4.

1.2.2 Ensemble référentiel

DÉFINITION 1.9 – On appelle ***(ensemble) référentiel,*** ou, parfois, ***univers du discours***, un ensemble U ou R (ou Ref) contenant tous ceux dont on parle (Venn, ~1860).

Pour raisonner sur des ensembles ayant un référentiel commun, on utilise un ***diagramme de Venn.*** Ce diagramme se présente comme un diagramme d'Euler *délimité par un rectangle* qui figure alors « l'univers du discours », *englobant tous les ensembles spécifiques.* La théorie se symétrise.

PROPOSITION 1.8 – Tout ensemble E est inclus dans le référentiel : $\varnothing \subseteq E \subseteq R$.

Le principe de compréhension (§ 1.1.5) permet de passer d'un ensemble au référentiel par une suite d'inclusions ; c'est en particulier le cas dans un formalisme objet, la classe la plus générale tenant lieu de référentiel, et toute sous-classe spécifique définissant un (sous-)ensemble inclus dans sa classe-mère.

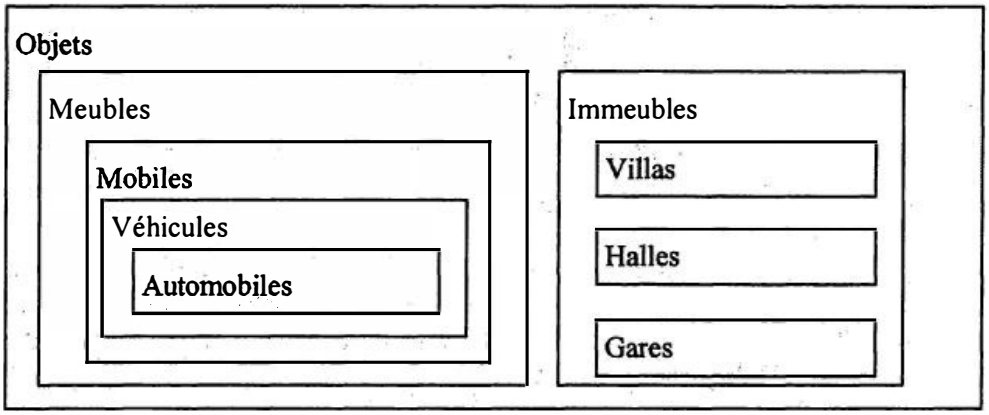


Fig. 1.3 Classes d'objets imbriquées par inclusion.

Cette descente du général au particulier s'interprète,

- en *extension*, comme une raréfaction progressive des individus ;
- en *compréhension*, comme un cumul de propriétés.

A la limite, une collection surabondante, voire contradictoire, de propriétés (n') est satisfaite (que) par 0 individu.

1.2.3 Inclusion stricte

DÉFINITION 1.10 – L'ensemble E est ***inclus au sens strict*** dans l'ensemble F, ou encore E est ***sous-ensemble strict*** de F

- si tout élément de E est élément de F,
- et si au moins un élément de F n'appartient pas à E.

Ce qu'on note : $E \subset F \equiv (E \subseteq F) \wedge (\exists x\ (x \in F) \wedge (x \notin E))$
$\equiv (E \subseteq F) \wedge \neg(F \subseteq E)$

Sinon, $E \not\subset F \equiv \neg(E \subset F)$

PROPOSITIONS

1.9 – pour tout ensemble E : $E \not\subset E$ /irréflexivité

1.10 – pour tous les ensembles E, F : $E \subset F \Rightarrow |E| < |F|$

1.11 – pour tous les ensembles E, F, G : $E \subset F, F \subset G \Rightarrow E \subset G$ /transitivité
$E \subset F \Rightarrow F \not\subset E$ /asymétrie

1.12 – $\subset$ est un ***ordre strict*** entre ensembles.

NOTE. Irréflexivité, asymétrie, ordre strict pointés ici seront définis au chapitre 4.

1.3 PARTIES D'UN ENSEMBLE

DÉFINITION 1.11 – On appelle ***ensemble des parties d'un ensemble*** E la collection $\mathcal{P}(E)$ de tous les sous-ensembles possibles de E.

Ce qu'on note : $\mathcal{P}(E) \equiv \{ A \mid A \subseteq E \}$

PROPOSITION 1.13 – Si un ensemble E possède n éléments, $\mathcal{P}(E)$ possède 2^n éléments. Ces derniers sont les sous-ensembles de E, dont exactement C^i_n à i éléments. Ainsi :

$$| \mathcal{P}(E) | = 2^{|E|}.$$

EXEMPLE. Soit E = {a, b, c, d} ; alors,

$\mathcal{P}$ (E) = {∅,	/ 1 ensemble à 0 élément
{a}, {b}, {c}, {d},	/ 4 ensembles à 1 élément
{a, b}, {a, c}, {a, d}, {b, c}, {b, d}, {c, d},	/ 6 ensembles à 2 éléments
{a, b, c}, {a, b, d}, {a, c, d}, {b, c, d},	/ 4 ensembles à 3 éléments
{a, b, c, d}	/ 1 ensemble à 4 éléments
}	Soit les $16 = 2^4$ sous-ensembles de E

NOTES

- Avec l'approche en compréhension, un sous-ensemble F était la réduction d'un ensemble E aux seuls éléments de E satisfaisant une propriété PF dite propriété caractéristique de F, d'où l'ensemble $F \equiv \{ x \in E \mid PF(x) \}$. Réciproquement, une propriété PF pourra être définie dans E par l'extension de l'ensemble $F \subseteq E$ des objets de E qui la satisfont, en posant :

$$P_F(x) \equiv x \in F$$

- Bien distinguer les *propriétés d'individus* des *propriétés d'ensemble.* Si A est l'ensemble des Anglais, et que l'on considère les propriétés :
  - « tout Anglais est mortel » : alors « être mortel » est une *propriété générique* pour tous les individus de A, qui peut être associée à A ; on note : $\forall x \in A$ mortel(x) ;
  - « des Anglais sont bruns » : alors il y en a au moins un, et « être brun » pour un individu est une *propriété spécifique ;* on note : $\exists x \in A$ brun(x) ;
  - « les Anglais sont roux » : en voulant seulement dire que les roux sont majoritaires parmi les Anglais, ou simplement qu'ils sont plus nombreux que les blonds ou les bruns, « être roux » pour un individu est alors une *propriété*

typique ; ce cas, important en matière de formalisation des connaissances empiriques, attend encore une formalisation logique adéquate.

Les énoncés du genre « les A sont B » seront donc ramenés selon les cas à
- « tous les A sont B », « chaque A est B », énoncés génériques, absolus, à l'usage bien formalisé,
- ou des énoncés typiques plus vagues comme « *la plupart* des A sont des B », « les A sont *le plus souvent* des B » ou « tous les A sont *plus ou moins* des B », d'exploitation nettement plus délicate.

1.4 UNION

DÉFINITION 1.12 – Soit C ={ X, Y, Z, ... } une collection d'ensembles. Leur ***union*** est l'ensemble formé des objets de chacun d'eux. On note:

$$\cup C \equiv \{ x \mid \exists A \in C \; x \in A \}$$

Si C = { X, Y}, on utilise la notation $X \cup Y$ plutôt que $\cup\{X,Y\}$.

Tableau 1.1 Union, définition logique.

$a \in X$	$a \in Y$	$a \in X \cup Y$
non	non	non
non	oui	*oui*
oui	non	*oui*
oui	oui	*oui*

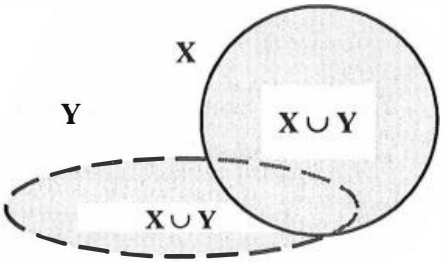


Fig. 1.4 Diagramme d'une union $X \cup Y$.

PROPOSITIONS

1.14 – Quels que soient les ensembles E, F, G, on a :

$E \cup (F \cup G) = (E \cup F) \cup G = \cup\{E, F, G\} = E \cup F \cup G$ /associativité[1]
$E \cup F = F \cup E$ /commutativité
$E \cup E = E$ /idempotence
$E \cup \varnothing = \varnothing \cup E = E$ /$\varnothing$ élément neutre

1.15 – Si les ensembles E et F sont finis :

$$\max(\,|E|, |F|\,) \leq |E \cup F| \leq |E| + |F|.$$

1.16 – Si on a un référentiel R : $E \cup F \subseteq R$, et si R est fini : $|E \cup F| \leq |R|$.

1.17 – $E \subseteq F \Leftrightarrow E \cup F = F$.

COROLLAIRE. Avec un référentiel R : $E \subseteq R \Leftrightarrow E \cup R = R$.

NOTE. On dit alors que le référentiel est un *élément absorbant* pour l'union.

[1] Ces propriétés (associativité, commutativité, idempotence, éléments neutres et absorbants...) seront définies au début de la partie Algèbre (chap. 12) ; pour une meilleure accoutumance, elles sont signalées au fur et à mesure des rencontres, sans exclure un rappel en illustration de leur définition.

1.5 INTERSECTION

1.5.1 Présentation ❦

DÉFINITION 1.13 – Soit C ={ X, Y, Z, ... } une collection d'ensembles. Leur ***intersection*** $\cap$ C est l'ensemble formé des *objets communs* à tous :

$$\cap C \equiv \{ x \mid \forall E \in C : x \in E \}$$

Pour deux ensembles on utilisera le plus souvent :

$$X \cap Y \equiv \cap\{X, Y\}$$

Tableau 1.2 Intersection.

$a \in X$	$a \in Y$	$a \in X \cap Y$
non	non	non
non	oui	non
oui	non	non
oui	oui	*oui*

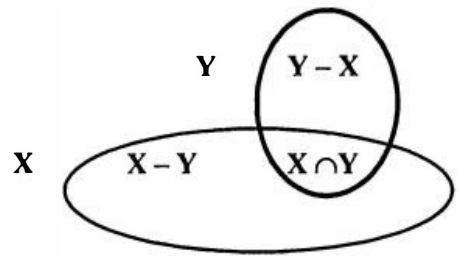


Fig. 1.5 Intersection et différences.

PROPOSITIONS

1.18 – Quels que soient les ensembles E, F, G :

$E \cap (F \cap G) = (E \cap F) \cap G = \cap \{E, F, G\} = E \cap F \cap G$ /associativité

$E \cap F = F \cap E$ /commutativité

$E \cap E = E$ /idempotence

$E \cap \varnothing = \varnothing$ /$\varnothing$ élément absorbant

1.19 – Pour les ensembles E, F finis :

$0 \leq |E \cap F| \leq \min(|E|, |F|)$,

$|E| + |F| = |E \cap F| + |E \cup F|$.

1.20 – Pour tous les ensembles E, F partageant un référentiel fini R :

$\max(0, |E| + |F| - |R|) \leq |E \cap F| \leq \min(|E|, |F|)$.

1.21 – $E \subseteq F \Leftrightarrow E \cap F = E$.

1.22 – Pour tous les sous-ensembles E d'un référentiel Ref :

$E \subseteq Ref \Leftrightarrow E \cap Ref = E$.

Le référentiel est alors dit *élément neutre de l'intersection.*

1.5.2 Ensembles disjoints

DÉFINITION 1.14 – Deux ensembles E, F sont disjoints si leur intersection est vide :

$$E \text{ et } F \textbf{ ***ensembles disjoints*** } \equiv E \cap F = \varnothing$$

REMARQUE. On note parfois additivement l'union d'ensembles disjoints :

$$E + F \equiv E \cup F \textit{ ssi } E \cap F = \varnothing$$

1.5.3 Distributivités

Soit C = {X, Y, Z, ... } une collection d'ensembles, et A un ensemble donné quelconque. On montre :

PROPOSITIONS

1.23 – $(\cup C) \cap A = \cup\{ X \cap A \mid X \in C \}$.

1.24 – $(\cap C) \cup A = \cap\{ X \cup A \mid X \in C \}$.

1.25 – Pour 3 ensembles A, B, C quelconques on a :

$$(B \cup C) \cap A = (B \cap A) \cup (C \cap A),$$
$$(B \cap C) \cup A = (B \cup A) \cap (B \cup C),$$

... et $\cup$ et $\cap$ sont dits *mutuellement distributifs.*

1.6 DIFFÉRENCE ENSEMBLISTE

1.6.1 Présentation

DÉFINITION 1.15 – Soient X, Y deux ensembles. Leur *différence* $X - Y$ est l'ensemble des objets de X qui n'appartiennent pas à Y :

$$X - Y \equiv \{ x \in X \mid x \notin Y \}$$

(voir diagramme figure 1.5, définition logique tableau 1.3).

Tableau 1.3 Différences, définition logique.

$a \in X$	$a \in Y$	$a \in X - Y$	$a \in Y - X$
non	non	non	non
non	oui	non	*oui*
oui	non	*oui*	non
oui	oui	non	non

1.6.2 Complément

DÉFINITION 1.16 – Lorsque, dans une différence A – B, A n'a plus à être nommé, étant par convention le *référentiel*, l'*univers du discours,* incluant toutes les combinaisons des ensembles en cause, on parle de ***complément*** de B (sous-entendu : *au référentiel A*), qu'on note B *surligné* ou *primé*

$$B' \equiv \{ x \in A_{ref} \mid x \notin B \} \equiv A_{ref} - B$$

On peut considérer une différence A – B comme le complément de B *relatif à* A pris comme référentiel.

1.6.3 Propriétés

PROPOSITION 1.26 – Pour tout ensemble E :

$\varnothing - E = \varnothing$ /$\varnothing$ élément absorbant à gauche

$E - \varnothing = E$ /$\varnothing$ élément neutre à droite

PROPOSITION 1.27 – Soient C = {X, Y, Z, ...} une collection d'ensembles, et A un ensemble quelconque.

$$A - (\cup C) = \cap\{ A - P \mid P \in C \}$$
$$A - (\cap C) = \cup\{ A - P \mid P \in C \}$$

COROLLAIRE 1.28 – Pour 3 ensembles A, B, C quelconques

$$A - (B \cup C) = (A - B) \cap (A - C)$$
$$A - (B \cap C) = (A - B) \cup (A - C)$$

soit, en termes de complément :

$$(B \cup C)' = B' \cap C'$$
$$(B \cap C)' = B' \cup C' \qquad \text{(G. d'Occam / A. de Morgan)}$$

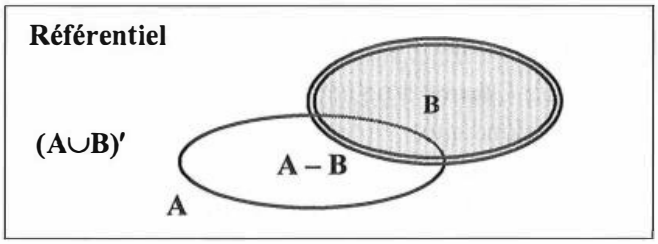


Fig. 1.6 Ensembles et complément (diagramme d'Euler-Venn).

1.7 COUVERTURES

1.7.1 Présentation

DÉFINITION 1.17 – Soit C = {x, y, z, ...} une *collection d'éléments* et Γ = {X, Y, Z, ...} une *collection de sous-ensembles* de C. Alors,

$$\Gamma \textit{ couvre } C \equiv C \subseteq \cup \Gamma$$

PROPOSITION 1.29 – Si Γ couvre C, $\forall x \in C\ (\exists A \in \Gamma\ (x \in A))$.

EXEMPLES

- Si C est un ensemble d'*individus* membres d'*associations*, une couverture de C est une liste d'associations telle que chaque individu soit membre d'au moins l'une d'elles.
- Si C est un ensemble *d'achats à faire*, on considérera comme couverture *une liste de magasins ou de fournisseurs* suffisante pour pouvoir tous les faire.

1.7.2 Finesse

DÉFINITION 1.18 – Soient Γ = {X, Y, Z...} et Γ* = {X*, Y*, Z*...} deux couvertures de C. On dira Γ* ***plus fine que*** Γ, et on notera $\Gamma^* \sqsubseteq \Gamma$ si tout ensemble de Γ* est inclus dans un ensemble de Γ. Soit :

$$\Gamma^* \sqsubseteq \Gamma \equiv \forall X^* \in \Gamma^*\ (\exists X \in \Gamma\ (X^* \subseteq X))$$

1.7.3 Redondance / irredondance

DÉFINITION 1.19 – Soit une couverture Γ et A un de ses ensembles ; A est dit ***ensemble redondant*** (pour Γ) s'il est inclus dans l'union des autres.

$$A \textbf{ \textit{redondant pour} } \Gamma \equiv A \subseteq \cup (\Gamma - \{A\})$$

Alors, aucun élément couvert ne lui est propre, et :

PROPOSITION 1.30 – Si A redondant pour Γ, $\cup (\Gamma - \{A\}) = \cup \Gamma$

DÉFINITION 1.20 – Une couverture Γ est dite ***couverture irredondante*** si aucun de ses ensembles n'est redondant pour elle.

PROPOSITION 1.31 – A toute couverture *finie* Γ correspond au moins une couverture irredondante plus fine que Γ.

PREUVE. Démonstration par descente récursive :

Toute couverture formée d'un ensemble non vide est irredondante.

Toute couverture Γ formé d'un nombre fini d'ensembles :

- soit est irredondante,
- soit comporte au moins un ensemble A redondant : alors, la couverture $\Gamma^* = \Gamma - \{A\}$ couvrira les mêmes points que Γ avec un ensemble de moins ; cette nouvelle couverture Γ^* à son tour, soit sera irredondante (par exemple parce qu'elle ne comporte plus qu'un ensemble), soit comportera au moins un ensemble redondant B...

A chaque étape, la nouvelle couverture formée comprend un ensemble de moins que la précédente, ce qui mène nécessairement à une couverture irredondante en moins de $|\Gamma|$ étapes. ■

NOTE. Dans de nombreux problèmes de couverture optimale, l'irredondance sera une condition nécessaire de l'optimalité.

1.7.4 Exemple

Soit C = {q, r, s, t, u, v, w, x, y, z }; F = {q, r}; G = {r, s, t}; H = {s, t, u, v}; I = {u, v, w}; J = {w, x}; K = {x, y, z} et Γ = {F, G, H, I, J, K}.

C = $\cup \Gamma$, et Γ est donc *couverture* de C.

Partant de Γ, quelles sont les couvertures irredondantes de C ?

Les données sont résumées par le tableau 1.4 ci-après, qui confronte les ensembles couvrants (1 par ligne) et les points couverts (1 par colonne). La dernière ligne indique par combien d'ensembles chaque point est couvert.

Observons les points couverts 1 fois : la couverture de q exige F et celle de y et z exige K ; F et K seront *nécessaires* à toute couverture de C.

A contrario, G, H, I, J sont des ensembles *redondants* de cette couverture, car chacun des points qu'ils couvrent est couvert au moins 2 fois.

Les couvertures irredondantes associées à Γ comprennent donc F et K, et de quoi couvrir C – F – K = { s, t, u, v, w, x}.

Pour couvrir s et t, il faut G ou H ; pour couvrir u et v, il faut H ou I ; enfin, w exige I ou J.

D'où 3 couvertures irredondantes :

Γ_1 = {F, *H, I*, K} ; Γ_2 ={F, *H, J*, K} ; Γ_3 = {F, *G, I*, K}.

Tableau 1.4 Ensembles couvrants et points couverts.

Points / Ensembles	q	r	*s*	*t*	*u*	*v*	*w*	x	y	z
F	⊗	x								
G		x	x	x						
H			x	x	x	x				
I					x	x	x			
J							x	x		
K								x	⊗	⊗
Nb. d'ensembles couvrants	1	2	2	2	2	2	2	2	1	1

⊗ : points critiques ; grisé : ensembles nécessaires ; rectangle : problème résiduel.

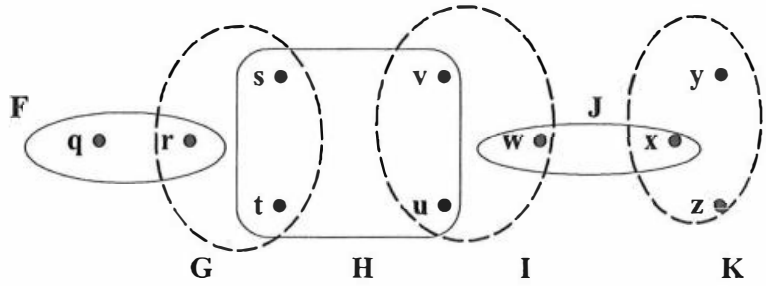


Fig. 1.7 Diagramme d'Euler de la couverture Γ.

1.8 DIFFÉRENCE SYMÉTRIQUE

1.8.1 Présentation

DÉFINITION 1.21 – La ***différence symétrique*** *de deux ensembles* A et B est l'ensemble noté A Δ B des objets appartenant *soit à A soit à B, mais pas aux deux* :

$$A \Delta B \equiv (A \cup B) - (A \cap B)$$

Tableau 1.5 Comparaison logique des différences, union, intersection.

a∈X	a∈Y	a∈X–Y	a∈Y–X	a∈XΔY	a∈X∪Y	a∈X∩Y
non	non	non	non	non	non	non
non	oui	non	*oui*	*oui*	*oui*	non
oui	non	*oui*	non	*oui*	*oui*	non
oui	oui	non	non	non	*oui*	*oui*

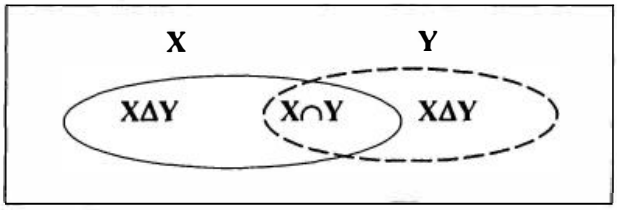


Fig. 1.8 Diagramme différence symétrique et intersection.

1.8.2 Propriétés

PROPOSITIONS

1.32 – Pour tous les ensembles A, B :

$$A \Delta B \equiv (A - B) \cup (B - A)$$
$$A \Delta B \equiv B \Delta A \qquad \text{/commutativité}$$
$$A \Delta B \equiv (A - B) \Delta (B - A)$$
$$A \cup B \equiv (A - B) \Delta (B - A) \Delta (A \cap B) \equiv A \Delta B \Delta (A \cap B)$$

1.33 – Pour tout ensemble E :

$$E \Delta \varnothing \equiv \varnothing \Delta E \equiv E \qquad \text{/}\varnothing \text{ élément neutre}$$
$$E \Delta E \equiv \varnothing$$

1.34 – Pour tous les ensembles A, B, C :

$$(A \Delta B) \Delta C = A \Delta (B \Delta C) = A \Delta B \Delta C \qquad \text{/associativité}$$
$$A \Delta B = A \Delta C \Rightarrow B = C \qquad \text{/simplifiabilité à gauche}$$
$$B \Delta A = C \Delta A \Rightarrow B = C \qquad \text{/simplifiabilité à droite}$$
$$A \Delta B = C \Rightarrow B = A \Delta C \qquad \text{/changement de membre}$$

1.35 – Avec un référentiel Ref : $\text{Ref} \Delta A = A'$

1.9 PARTITIONS

1.9.1 Présentation

DÉFINITION 1.22 – On appelle ***partition d'un ensemble*** E, une couverture de E par une collection d'ensembles disjoints deux à deux.

Une partition peut être *plus fine* qu'une autre. La plus fine des partitions est la ***partition discrète*** qui isole chaque élément de E ; la moins fine des partitions est la ***partition universelle*** {E} qui rassemble tous les éléments de E dans leur ensemble commun.

Si chaque élément d'un ensemble E fini a reçu *un* numéro, on forme une partition P de E par la collection des E_n, ensemble des éléments de E de numéro n. Cette partition P est la plus fine possible si chaque élément de E a reçu un numéro *distinct*.

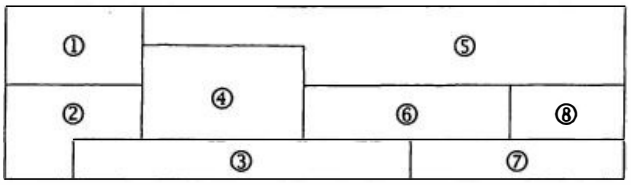


Fig. 1.9 Partition d'un ensemble E en 8 sous-ensembles.

1.9.2 Partitions & couvertures

PROPOSITIONS

1.36 – Toute partition est une couverture irredondante.

1.37 – A toute couverture finie, on peut associer une partition plus fine (au sens du § 1.7.2).

PREUVE. Soit C une couverture d'un ensemble X.

- Si elle ne comporte qu'un ensemble, c'est une partition (la moins fine).
- Si elle comporte au moins deux ensembles :
  - s'ils sont tous disjoints deux à deux, c'est une partition ;
  - sinon, soient F et G deux ensembles non disjoints de C ; on les remplace par les 2 ensembles disjoints F et G – F, qui couvrent les mêmes points ; on obtient ainsi une couverture C+ strictement plus fine que C ; ou C+ est une partition, ou il existe dans C+ deux ensembles F+ et G+ non disjoints...

Le processus est convergent pour un ensemble fini X, car le nombre de ses couvertures est fini, la couverture la plus fine d'un ensemble X étant précisément sa partition discrète. Le processus peut être appliqué soit à la couverture à traiter, soit à une couverture irredondante associée. ■

EXEMPLE. Reprenons le cas de la couverture traitée précédemment.

Soit $C = \{q,r,s,t,u,v,w,x,y,z\}$; F = {q, r}; G = {r, s, t}; H = {s, t, u, v}; I = {u, v, w}; J = {w, x}; K = {x, y, z}; et $\Gamma = \{F, G, H, I, J, K\}$ une couverture associée.

Quelle partition plus fine peut-on dériver de Γ?

Dans Γ, F et G ne sont pas disjoints, G – F = {s, t}.

On pose Γ_1 = {{q, r},{s,t}, H, I, J, K} ; {s, t} et H = {s, t, u, v} ne sont pas disjoints ; H – {s, t} = {u, v} ;

On pose Γ_2 = {{q, r}, {s,t}, {u,v}, I, J, K} ; {u, v} et I = {u, v, w} ne sont pas disjoints ; I – {u, v} = {w} ;

On pose Γ_3 = {{q, r},{s,t}, {u,v}, {w}, J, K} ; {w} et J ne sont pas disjoints. On forme Γ_4 = {{q, r},{s,t}, {u,v}, {w}, {x}, K} ; en considérant le couple ({x}, K) on obtient enfin la partition :

$$\Pi = \{\{q, r\}, \{s,t\}, \{u,v\}, \{w\}, \{x\}, \{y,z\}\}$$

qui satisfait $\Pi \subseteq \Gamma_4 \subseteq \Gamma_3 \subseteq \Gamma_2 \subseteq \Gamma_1 \subseteq \Gamma$.

En partant de la couverture irredondante Γ^+ = {F, H, I, K} on obtient plus rapidement

$$\Pi^+ = \{\{q, r\}, \{s,t, u,v\}, \{w\}, \{x, y,z\}\},$$

partition plus proche de Γ puisque $\Pi \subseteq \Pi^+ \subseteq \Gamma$.

1.9.3 Finesse entre partitions

La finesse entre couvertures peut être restreinte à une ***finesse entre partitions***.

On appelle ***chaîne de finesse*** une liste de partitions dont chacune est plus fine que la suivante.

Une chaîne de finesse entre partitions d'un même ensemble s'interprète comme l'observation d'une même *classification arborescente* à différents niveaux de finesse, de la partition la plus fine, qui isole chaque élément, à la partition la plus grossière, qui rassemble tous les éléments en une même classe.

Tableau 1.6 Partitions successives d'une classification arborescente.

niveau								
0	(abcdefgh)							
1	(abcde)					(fgh)		
2	(abcde)					(f)	(gh)	
3	(ab)		(cde)			(f)	(gh)	
4	(a)	(b)	(cde)			(f)	(g)	(h)
5	(a)	(b)	(c)	(de)		(f)	(g)	(h)
6	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)

Chaîne de finesse d'une classification arborescente

Le tableau 1.6 nous donne

$P_6 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}\}$ = *« partition la plus fine »*
$\subseteq$ $P_5 = \{\{a\}, \{b\}, \{c\}, \{d, e\}, \{f\}, \{g\}, \{h\}\}$
$\subseteq$ $P_4 = \{\{a\}, \{b\}, \{c, d, e\}, \{f\}, \{g\}, \{h\}\}$
$\subseteq$ $P_3 = \{\{a, b\}, \{c, d, e\}, \{f\}, \{g, h\}\}$
$\subseteq$ $P_2 = \{\{a, b, c, d, e\}, \{f\}, \{g, h\}\}$
$\subseteq$ $P_1 = \{\{a, b, c, d, e\}, \{f, g, h\}\}$
$\subseteq$ $P_0 = \{\{a, b, c, d, e, f, g, h\}$ = *« partition la moins fine»*.

La suite P_6, P_5, P_4, P_3, P_2, P_1, P_0 est dite *chaîne de finesse* pour ces partitions : c'est une séquence totalement ordonnée, de la plus fine, ou *partition discrète* à la moins fine, ou *partition universelle* (fig. 1.10).

1.10 PAIRES & PRODUITS CARTÉSIENS

1.10.1 Paires ordonnées

Par définition des extensions, $\{a, b\} = \{b, a\}$, et $\{a, a\} = \{a\}$. Par opposition,

DÉFINITION 1.23 – On appellera ***paire ordonnée*** des couples notés (a, b) où le rang (1er pour a, 2nd pour b) est imposé et significatif[2].

POSTULAT 1.38 – (a, b) = (c, d) *si et seulement si* a = b et c = d.

[2] Certains auteurs posent $(a, b) \equiv \{a, \{a, b\}\}$, pour décrire précisément (a, b) comme une *paire {a, b} dont un élément a est distingué.*

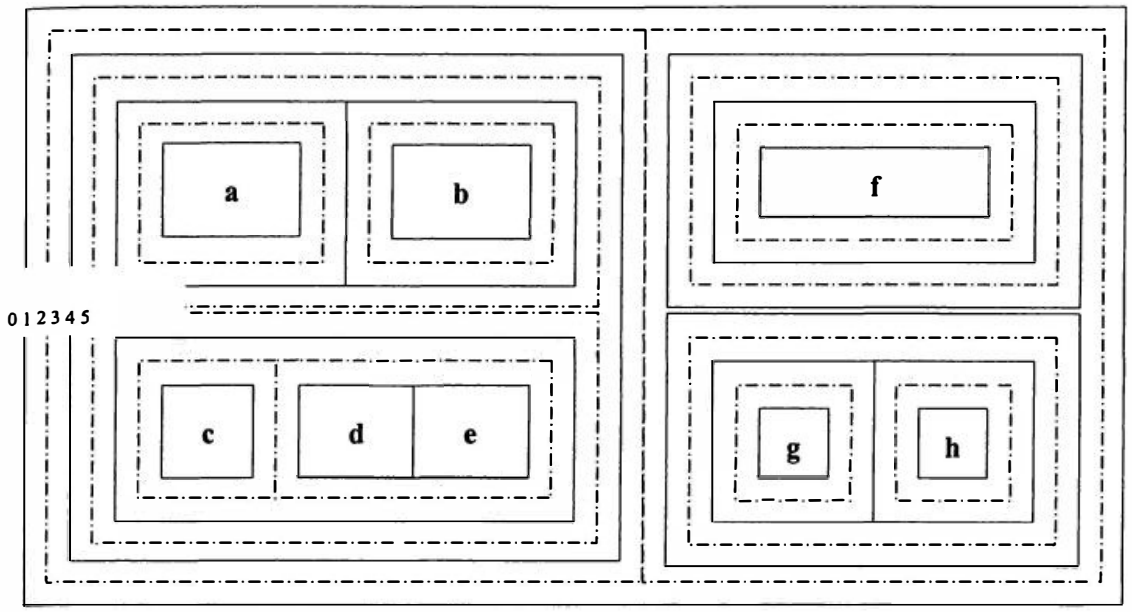


Fig. 1.10 Partitions successives d'une classification arborescente, diagramme ensembliste.

1.10.2 Produit cartésien

DÉFINITION 1.24 – Soient A, B deux ensembles. Leur ***produit cartésien*** est l'ensemble des paires ordonnées formées d'un objet de A et d'un objet de B. On note :

$$A \otimes B = \{(a, b) \mid a \in A \wedge b \in B\}$$

EXEMPLE. {a1, a2}⊗{b1, b2, b3} = {(a1, b1), (a1, b2), (a1, b3), (a2, b1), (a2, b2), (a2, b3)}. Cet ensemble est symbolisé par les cases d'un tableau rectangulaire dont les lignes sont relatives aux éléments de A et les colonnes à ceux de B (tab. 1.7).

Tableau 1.7 Produit cartésien A⊗B.

A \ B	b1	b2	b3
a1	(a1, b1)	(a1, b2)	(a1, b3)
a2	(a2, b1)	(a2, b2)	(a2, b3)

PROPOSITIONS

1.39 – $|A \otimes B| = |A| \cdot |B|$

1.40 – Soient 4 ensembles A, B, C, D quelconques :

$A \subseteq B,\ C \subseteq D \Rightarrow A \otimes C \subseteq B \otimes D$ /monotonie

1.11 EXERCICES

1.11.0

Compléter le tableau 1.8 (au crayon d'abord !).

Tableau 1.8 Points et ensembles.

Points \ Ensembles	A	B	Ref	A∩B	A–B	B–A	AΔB	A∪B	A'	B'	A'∩B'	(A∪B)'
a	x		x									
b	x	x	x									
c		x	x									
d		x	x									
e			x									
f			x									

1.11.1

Démontrer les lois d'absorption : $A \cup (A \cap B) = A = A \cap (A \cup B)$.

1.11.2

Montrer que si A et B sont disjoints, alors $A \Delta B = A \cup B$; s'en servir pour établir la proposition 1.32 quater.

1.11.3

Montrer que pour 3 ensembles :

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

1.11.4

Montrer que le nombre de couvertures d'un ensemble X à n éléments est fini.

1.11.5

Définir en fonction de B et C, A tel que $A \cup B = A \cap C$.

1.11.6

Soit {A, B, C, ..., K} une collection d'ensembles de référentiel R. Montrer que tous les ensembles constructibles à partir de cette collection et des opérateurs $\cup$, $\cap$, $'$, Δ le sont à partir de la seule différence ensembliste $-$.

1.11.7

Dresser un tableau à 2 dimensions mettant en regard des langues officielles correspondantes,

- les pays de l'Europe des Six,
- ceux de l'Union Européenne actuelle.

Peut-on dire à ces deux stades que la liste des langues officielles de l'Europe couvre la liste des langues officielles ces pays ?

Montrer qu'une langue est *obligatoire* (pour l'Europe) si elle est langue officielle d'un pays membre unilingue. Un pays est-il *critique* dès lors qu'il est unilingue ?

30% du budget de fonctionnement administratif de l'Union Européenne est absorbé par les traductions ; la langue devrait-elle être un critère d'admission ? Quel sera(it) l'effet de l'admission des pays de Visegrad ? de la Suisse? de la Norvège ? ...

1.11.8 Théorie additive des couleurs

A partir des 3 couleurs primaires : rouge magenta, vert, bleu, expliquer comment former 8 couleurs distinctes.

Présenter le résultat sous forme d'un diagramme d'Euler-Venn.

1.12 PROBLÈMES

1.12.0

Le club Sport & Culture comporte 230 membres qui paient exactement leur cotisations. Cette cotisation est de 100 francs pour une activité, 150 francs pour deux activités. Les 2 responsables d'activités comptent respectivement 145 et 105 membres. Quelle masse de cotisations peut espérer le Bureau du Club ?

1.12.1

Même problème en supposant maintenant que la cotisation de 150 francs est pour 2 activités *ou plus,* et que les 3 responsables d'activités annoncent respectivement 100, 70 et 90 membres. Quelle masse de cotisations peut espérer le Bureau du Club ? Quel est l'écart entre les 2 prévisions *extrêmes?*

1.12.2

Un observateur constate que, dans cette société isolée, il existe différentes corporations, fort jalouses les unes des autres, qui dénoncent vigoureusement leurs privilèges respectifs, semble-t-il sans mentir.

Sans comprendre tous les détails, il voit que les A reprochent aux B 3 privilèges ; que les B reprochent aux C 4 privilèges, que les C reprochent aux D 5 privilèges, et que ceux-ci en reprochent 6 aux A.

Avec combien de privilèges différents p1, p2, p3... peut-on expliquer ce système social :

- si les privilèges de noms différents sont différents,
- si un même privilège peut être nommé différemment par 2 corporations.

Montrer qu'un privilège partagé se traduit ici par un non-dit, et que seuls sont repérables dans un tel système les privilèges qui font problème à une partie de la population.

(Adapté d'un problème de biométrie de J. L. Sourrouille.)

1.12.3 Reconstitution d'archives

Le jour où il ne faut pas, vous découvrez que :

- vous avez besoin du fichier clients C, et du fichier prospects P qui contenait la liste des clients prospectés, c.à.d. des clients actuels ou potentiels visités par les représentants au dernier trimestre ;
- le stagiaire les a effacés par mégarde, en répondant au hasard à une question du système qu'il ne comprenait pas.

Au cours d'une réunion de crise, vous apprenez cependant qu'il reste :

- le fichier F des clients non prospectés ce dernier trimestre,

- le fichier G des prospectés du dernier trimestre non encore clients,
- le fichier H des clients et/ou prospects, mélangés sans distinction.

Exprimer F, G, H en fonction de P et C.
Exprimer P et C en fonction de F, G et H.
En déduire comment reconstruire P et C.

1.12.4 Fournisseurs et fournitures

A l'aube, le prototype qui devrait être présenté demain a été accidenté. Il faut d'urgence des pièces a, b, c, d, e, f , g, h qui ne sont pas en stock. Après enquête auprès de vos fournisseurs sur les fournitures disponibles chez eux, on relève que :

A	peut	fournir	a, b
B	--	--	b, c, d
C	--	--	d, e, f, g, h
D	--	--	h, i, j, a
E	--	--	b, d, h, a.

La circulation est fluide mais le stationnement atroce, et le prix sans importance ; chez qui suffit-il d'aller pour acquérir une collection {a, b, c, d, e, f, g, h}?

On procèdera de deux façons différentes.

On dressera une table rectangulaire fournisseurs / fourniture ; on cherchera les fournitures *critiques* (à 1 seul fournisseur), les fournisseurs *nécessaires,* l'ensemble des fournitures que ces derniers peuvent fournir, celles restant à trouver...

On assignera à chaque fourniture l'union de ses fournisseurs, et on fera l'intersection de ces ensembles. En développant cette intersection d'unions en union d'intersections, et en utilisant la propriété d'absorption, on montrera que l'on retrouve les résultats précédents.

1.12.5 Langues & communication

Le Secrétariat de la Commission Technique pour l'Informatique de la Société des Nations doit assurer la communication rapide d'une série de documents à chaque membre de la Commission. On décide de tirer ces documents dans un nombre minimal de langues, *de telle façon que chaque membre puisse en prendre connaissance dans la langue ou l'une des langues qu'il a indiquée(s) au préalable.* L'enquête préalable donne la table 1.9 suivante (on a déjà enlevé les langues parlées par un seul expert, et récusé les experts ne parlant que 0 ou 1 des langues restantes).

Chercher les inclusions du style *Dupont* $\subseteq$ *Gladstone,* où *Dupont* désigne l'ensemble des langues admises par Dupont.

Chercher les inclusions du style *turc* $\subseteq$ *russe*, où *turc* désigne l'ensemble des experts admettant le turc.

Définir les lignes et colonnes qu'on peut retirer sans modifier la/les solution(s) finale(s).

Dresser un tableau simplifié.

Proposer une ou des listes finales.

Interpréter les retraits de lignes et de colonnes en termes de redondance, criticité...

Tableau 1.9 Langues pratiquées par les divers experts.

Langue / Expert	Chinois	Hindi	Russe	Turc	Anglais	Espagnol	Portugais	Japonais	Allemand	Français
Y.Tchourmine	x		x	x						
P. Dupont					x					x
I. Gladstone		x			x					x
J. Navarro						x			x	x
K. Suzuki	x		x					x		
F. Da Silva						x	x			
R. Tagore		x			x		x			
Lee Zefong	x		x					x		
C. van Meulen			x		x				x	x
K. Cimoglu			x	x					x	
Zin Tekwan	x	x			x			x		

1.12.6 Langues & communautés

La Mairie Centrale de Mégapolis veut placarder rapidement des affiches avertissant tous les habitants des précautions à prendre contre une violente épidémie qui se rapproche dangereusement.

Les habitants sont d'origines et de cultures variées, et forment un grand nombre de communautés caractérisées par une ou plusieurs langues.

On décide de tirer les affiches dans un nombre minimal de langues, de telle façon que chaque habitant puisse en prendre connaissance dans l'une des langues de la communauté (ou des communautés) à laquelle (auxquelles) il appartient. L'enquête préalable donne le tableau 1.10.

Chercher la collection N des langues *nécessaires* (i.e. utilisées par une ou plusieurs communautés unilingues).

Déterminer la liste des communautés couvertes par N.

Parmi les communautés restantes, en considérer une parlant 2 langues (A, B). Progresser en considérant successivement $N \cup \{A\}$ puis $N \cup \{B\}$ comme couvertures partielles...

Donner les couvertures finales les plus simples possibles.

Appelons *langue véhiculaire* celle parlée par le plus grand nombre de communautés ; est-il utile de la considérer ?

Tableau 1.10 Langues et Communautés.

Langue / Communauté	Chinois	Hindi	Turc	Espagnol	Portugais	Japonais	Allemand	Français
Soyeux	x		x					
Auvergnats								x
Mauriciens		x						x
Fils du Condor				x			x	x
Kamikazés	x					x		
Ibères				x	x			
Goans		x			x			
Formosans	x				x	x		
Luxembourgeois							x	x

1.12.7

Une association comprend un grand nombre de membres inscrits dans un ou plusieurs *collèges* correspondant aux secteurs d'activités de l'association.

Montrer que la collection C de ces collèges c_i constitue une couverture des effectifs de l'association, dès lors que tout membre doit être inscrit dans au moins un collège.

L'association désire émettre des circulaires, en fonction du *profil* de chaque membre (i.e. du ou des collèges auquel il appartient). Montrer que la répartition des membres en *groupes de même profil* constitue une partition.

Finalement, le nombre de collèges est devenu trop élevé pour le budget de l'association. Que se passera-t-il si on ne supprime que les collèges n'ayant aucun membre en propre ?

1.12.8 Algorithmie

On appelle *algorithme-glouton* un algorithme progressant sans retour en arrière. Donner un tel algorithme glouton pour la transformation d'une couverture en une partition plus fine.

1.13 LECTURES

AHO A., ULLMAN J., *Concepts fondamentaux de l'informatique*, chap. VII, Dunod, 1993.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science, vol. I,* chap. I, Coll. Texts and Monographs on Computer Science, Springer Verlag.

KUNTZMANN J., *Algèbre de Boole*, Dunod, Paris, 1969.

REINHARDT F. & SOEDER H., *Atlas des mathématiques*, chap. 2, La Pochothèque, Le Livre de Poche, Paris, 1997.

VÉLU J., *Méthodes mathématiques pour l'informatique*, chap. 1 à 3, Dunod, 1994.

Collectif, *Petite encyclopédie des mathématiques*, chap. 14, Edition Pagoulatos, Paris, 1980.

CHAPITRE 2

RELATIONS BINAIRES

Elles apparaissent dès qu'il y a une correspondance entre éléments d'un ensemble A et éléments d'un ensemble B.

2.1 VUE EN EXTENSION

2.1.1 Présentation

DÉFINITION 2.1 – Une ***relation binaire*** R de A à B est une *partie* de A⊗B :

$$R, \text{ relation de A à B} \equiv R \subseteq A \otimes B$$

On dit la ***relation interne*** si $B \subseteq A$, ***externe*** sinon.

On notera éventuellement R(x, y) ou x R y le prédicat $(x,y) \in R$ signifiant que « le couple (x,y) satisfait la relation R ».

Dans un tel couple, x sera dit ***origine***, et y ***image.***

EXEMPLES

- V ensemble de Villes, R ensemble de Régions, estEn = {(v, r)∈V⊗R| v estEn r}, *relation externe.*
- Pour P, ensemble de Personnes,
  pereOuMereDe = {(p1, p2) ∈P ⊗ P | p1 pèreOuMèreDe p2 }, *relation interne* à P.
- Un *dictionnaire* définit une *relation externe entre un ensemble d'entrées et un ensemble de définitions.*
- Entre personnes et langues, les relations externes « lit », « écrit », « parle » collecteront les énoncés *x lit L, x écrit L, x parle L.*

2.1.2 Représentations

Il s'agit de procédés de définition concrète des relations, qu'on peut considérer comme spécifiant des structures de données, dont le coût sera abordé au chapitre 6.

Liste de couples

Cette représentation calque strictement la définition en extension.

EXEMPLE. R = {(a, b) (a, c) (b, c) (b, d) (c, d) (e, a)}.

Table booléenne ou matrice caractéristique

A une relation $R \subseteq A \otimes B$ on associe une matrice booléenne [R] ainsi formée. Les éléments de A et de B étant numérotés :

- [R] comporte une i-ème ligne par élément $a_i \in A$,
- [R] comporte une j-ième colonne par élément $b_j \in B$,
- chaque élément r_{ij} de [R] a pour valeur r_{ij} = *(si* $(a_i, b_j) \in R$ *alors 1 sinon 0).*

Tableau 2.1 Matrice caractéristique d'une relation R.

[R]	b1	b2	b3	b4	b5
a1	0	0	0	1	1
a2	0	1	0	1	0
a3	1	0	1	0	0

Vecteur de vérité

L'ensemble des $(a_i, b_j) \in A \otimes B$ *étant supposé énuméré dans un ordre fixe*, à une relation $R \subseteq A \otimes B$ on associe le ***vecteur booléen*** R formé à raison d'un booléen par couple (a_i, b_j).

Liste de coupes

Cette liste associe à chaque élément a de A la liste de ses *images* ou *successeurs* dans B : R = {(a, b) (a, c) (b, c) (b, d) (c, d) (e, a)} devient R = { (a : b, c) (b : c, d) (c : d) (d : ∅) (e : a)}.

Table de successeurs

C'est une mise en table de la représentation précédente, utilisant 1 ligne par élément de A, et k colonnes pour ses successeurs, où k borne le nombre de points (de B) images d'un point (de A).

Diagramme sagittal ou graphe

Les éléments des ensembles deviennent des ***points***. A chaque couple ou paire ordonnée de la relation est associé un ***arc*** ou ***flèche***, reliant un point de A (***origine***) à un point de B (***image***) (voir aussi Partie II : les Graphes).

Tableau 2.2 Matrice caractéristique de la relation *<moteur> montéSur <modèle>*.

Modèles / Moteurs	141 L	141 N	141 LX	141 S	343 L	343 N	343 LX	343 S	545 L	545 N	545 LX	545 S
Z1	x											
Z2		x	x	x	x							
Z3						x	x		x			
Z4								x		x	x	
Z5												x

Notation intuitive : x pour 1, rien pour 0.

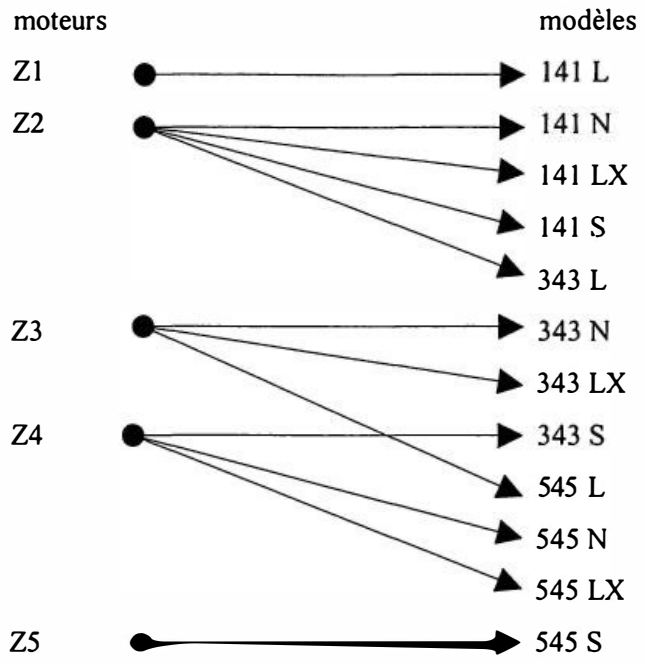


Fig. 2.1 Diagramme sagittal de la relation *<moteur> montéSur <modèle>*.

2.1.3 Domaine & image

Soit R, relation de A à B.

DÉFINITIONS

2.2 – On appelle ***domaine*** de R, l'ensemble des éléments de A ayant par R une image dans B. On note :

$$\mathrm{dom}(R) \equiv \{\, a \in A \mid \exists\, b \in B \;\; aRb \,\}$$

2.3 – On appelle ***image d'un point*** *a de A* ***par R***, l'ensemble éventuellement vide des éléments b de B tels que (a, b) $\in$R. On note :

$$\mathrm{im}_R(a) \equiv \{\, b \in B \mid (a, b) \in R \,\}$$

2.4 – On appelle ***image*** ou ***codomaine*** de R, l'ensemble des éléments de B image(s) par R d'au moins un élément de A. On note :

$$\mathrm{im}(R) \equiv \{ b \in B \mid \exists\, a \in A \; (a, b) \in R \}$$

EXEMPLE. Soit A = B = P, où P est un ensemble de Personnes, et la relation E = « être enfant de »

- dom(E) est le sous-ensemble des enfants, c'est-à-dire des personnes de P ayant au moins un parent dans P,
- imE(x) est l'ensemble des parents de x,
- im(E) est le sous-ensemble des parents, c'est-à-dire des personnes de P ayant au moins un enfant dans P.

PROPOSITIONS. Soit une relation $R \subseteq A \otimes B$.

2.1 – $\mathrm{dom}(R) \subseteq A$, $\mathrm{im}(R) \subseteq B$, $R \subseteq \mathrm{dom}(R) \otimes \mathrm{im}(R)$.

2.2 – $R \subseteq S \subseteq A \otimes B \Rightarrow \mathrm{dom}(R) \subseteq \mathrm{dom}(S)$, $\mathrm{im}(R) \subseteq \mathrm{im}(S)$.

2.1.4 Image et transformation

A la relation R on associe souvent la ***transformation*** Γ_R, définie par

$$\forall a \in A \ \Gamma_R a \equiv \{ b \mid (a, b) \in R \}$$

puis

$$\Gamma_R \equiv \{ (a, \Gamma_R a) \mid a \in A \}.$$

Γ_R a n'est donc qu'une autre notation pour $im_R(a)$, réduite même à Γa s'il n'y a pas d'ambiguïté sur la relation.

2.2 VUE EN COMPRÉHENSION

2.2.1 Présentation

L'approche en compréhension des ensembles, qui se réfère à leur signification, s'étend facilement à une relation définie en termes de prédicats.

DÉFINITION 2.5 – Une *relation binaire* R de A à B est un prédicat R(x, y) défini pour tout $x \in A$ et tout $y \in B$, auquel s'associe l'ensemble $R = \{(x, y) \in A \otimes B \mid R(x, y)\}$.

De fait, nous avons utilisé cette approche dans chaque exemple de relation s'appuyant sur une signification, par opposition à une liste arbitraire de paires de symboles. De ce point de vue, un même exemple peut être, pour le lecteur, en compréhension si l'exemple signifie pour lui quelque chose (il connaît les villes...), en extension sinon (il considère les noms comme des symboles sans signification).

La dualité de point de vue peut être maintenue dans le cas d'une *relation construite à partir d'autres,* la nouvelle relation étant vue

- soit comme ensemble, combinaison ensembliste des relations composantes,
- soit comme énoncé obtenu par composition logique des énoncés ou prédicats des relations composantes.

2.2.2 Ensemble de relations

DÉFINITION 2.6 – On notera $\mathfrak{R}_{AB}$ l'***ensemble des relations*** de A à B, de fait identique à $\mathcal{P}(A \otimes B)$. Dans un tel ensemble on pourra combiner des relations, les comparer...

2.2.3 Relations extrêmes

Nous bornons les ensembles de relations à l'aide de relations extrêmes, qui le sont dès à présent au sens de la logique, et le seront à partir de la section 2.4 au sens de la finesse.

DÉFINITION 2.7 – Soient A, B deux ensembles. Parmi les relations de A à B, on note :

- $\mathbf{0}_{AB}$ la ***relation nulle ou vide***, *jamais satisfaite*, dont la matrice caractéristique ne contiendra que des 0,
- $\mathbf{U}_{AB}$ la ***relation universelle ou complète***, *toujours satisfaite*, dont la matrice caractéristique ne contiendra que des 1.

2.3 RELATION INVERSE

2.3.1 Présentation

DÉFINITION 2.8 – La ***relation inverse***, notée R^{-1}, d'une relation R, est la collection de toutes les paires (b, a) telles que (a, b) satisfasse R. Ainsi :

$$R^{-1} \equiv \{(b, a) \mid (a, b) \in R\}$$

EXEMPLES

- Soit V un ensemble de Villes, R un ensemble de Régions, $E \subseteq V \otimes R$ la relation *(être) en* . La relation E^{-1} associera à une région ses villes : on la lira *contient* :

  saintEtienne *en* rhôneAlpes ≡ rhôneAlpes *en*$^{-1}$ saintEtienne
  ≡ rhôneAlpes *contient* saintEtienne

- Considérons un dictionnaire, ses entrées, ses définitions : l'inverse de la relation *<entrée>signifie<définition>* est une relation *<définition>définit<entrée>*, qui à chaque définition associe l'entrée (resp. les entrées synonymes) correspondante(s).

- L'inverse d'une relation *<personne> membreDe <club>* est la relation *<club> aPourMembre <personne>*.

2.3.2 Représentations associées

Graphe (d'une relation) inverse

Le graphe G^{-1} de la relation R^{-1}

- a pour sommets les sommets du graphe G de la relation R,
- a autant d'arcs (x y) que G possède d'arcs (y x), i.e. tous ses arcs sont obtenus *en inversant le sens des arcs d'origine.*

Matrice caractéristique d'une relation inverse

On passe de R à R^{-1} en permutant les rôles des ensembles de départ A et d'arrivée B : A, qui était lié aux lignes l'est maintenant aux colonnes, l'inverse pour B. Ainsi, $[R^{-1}]$, matrice de la relation inverse, est la *transposée* de la matrice booléenne [R] caractéristique de la relation d'origine.

Listes

Elles sont inversées, au sens de la construction de listes associées aux graphes inverses :

$$R = \{(a : b, c)\ (b : c, d)\ (c : d)\ (d : \varnothing)\ (e : a)\} \Rightarrow$$
$$R^{-1} = \{(a : e), (b : a), (c : a, b), (d : c), (e : \varnothing)\}$$

2.3.3 Propriétés

PROPOSITIONS. Soit une relation $R \subseteq A \otimes B$.

2.3 – *L'inverse R^{-1} est toujours défini*

$\mathrm{dom}(R^{-1}) = \mathrm{im}(R)$; $\mathrm{im}(R^{-1}) = \mathrm{dom}(R)$.

2.4 – $R \subseteq A \otimes B \leftrightarrow R^{-1} \subseteq B \otimes A$.

2.5 – $[R^{-1}] = [R]^T$.

2.6 – $(R^{-1})^{-1} = R$ /involution

2.7 – $R \subseteq S \subseteq A \otimes B \Rightarrow R^{-1} \subseteq S^{-1} \subseteq B \otimes A$ /monotonie

2.4 UNION, INTERSECTION, FINESSE

L'***union***, l'***intersection de relations*** dans $\mathfrak{R}_{AB}$ sont identiques à l'*union*, l'*intersection* des sous-ensembles correspondants de $A \otimes B$.

La ***finesse*** entre relations de $\mathfrak{R}_{AB}$ est identique à l'*inclusion* entre sous-ensembles correspondants de $A \otimes B$.

PROPOSITIONS. Soient $R, S \subseteq A \otimes B$ deux relations quelconques ; on a :

2.8 – $[R \cup S] = [R] + [S]$ i.e. $\forall i,j : [R \cup S]_{ij} = r_{ij} \vee s_{ij}$, où $\vee$ dénote le *ou inclusif*.

2.9 – $[R \cap S] = [R] \wedge [S]$ i.e. $\forall i,j : [R \cap S]_{ij} = r_{ij} \wedge s_{ij}$, où $\wedge$ dénote le *et*.

2.10 – $0_{AB} \cup R = U_{AB} \cap R = R = R \cup 0_{AB} = R \cap U_{AB}$.

2.11 – $0_{AB} \subseteq R \cap S \subseteq R \subseteq R \cup S \subseteq U_{AB}$.

NOTE. On verra au chapitre 4 que l'ensemble $\mathfrak{R}_{AB}$ des relations d'un ensemble A vers un ensemble B est ainsi doté par $\subseteq$, $\cup$, $\cap$ d'une structure ordonnée par $\subseteq$ dite *structure de treillis*, de minimum 0_{AB} et de maximum U_{AB}.

2.5 COMPOSITION

2.5.1 Présentation

DÉFINITION 2.9 – Soient les relations $R \subseteq A \otimes B$ et $S \subseteq B \otimes C$: on appelle ***composée*** de R et S la relation $T \subseteq A \otimes C$:

$$T \equiv \{ (a, c) \in A \otimes C \mid \exists b \in B \ (a \ R \ b \wedge b \ S \ c) \}$$

ce qu'on notera $T = R \bullet S$.

EXEMPLES

- Soit P l'ensemble des personnes travaillant sur un projet, E l'ensemble des équipes qu'elles forment, A l'ensemble des activités du projet. Soit $R \subseteq P \otimes E$ la relation exprimant si telle *personne* appartient à telle *équipe*. Soit $S \subseteq E \otimes A$ la relation exprimant si telle *équipe* est ou sera affectée à telle *tâche ou activité*. La relation composée $T = R \bullet S \subseteq P \otimes A$ exprime si telle *personne* pourra se retrouver affectée à telle *activité* du projet, au titre d'une équipe ou d'une autre.
- Soit Π un ensemble de personnes, P la relation exprimant que telle personne est père de telle autre, M la relation exprimant que telle personne est mère de telle autre. La relation composée $G = P \bullet M$ est satisfaite s'il existe un couple (x y) de personnes telles qu'il existe un(e) z dont x soit le père, et qui soit mère de y, i.e. ssi *x est grand-père maternel de y*.

PROPOSITIONS. Soient une relation $R \subseteq A \otimes B$ et une relation $S \subseteq B \otimes C$; on a :

2.12 – $\mathrm{dom}(R \bullet S) \subseteq \mathrm{dom}(R)$, $\mathrm{im}(R \bullet S) \subseteq \mathrm{im}(S)$.

2.13 – $0_{AB} \bullet S = 0_{AC} = R \bullet 0_{BC}$.

2.14 – Si $T \subseteq C \otimes D$, $R \bullet (S \bullet T) = (R \bullet S) \bullet T$ /associativité

2.15 – Si $T \subseteq B \otimes C$, $R \bullet (S \cup T) = (R \bullet S) \cup (R \bullet T)$ /distributivité $\bullet / \cup$

2.16 – $(R \bullet S)^{-1} = S^{-1} \bullet R^{-1}$

PREUVE. $c(R \bullet S)^{-1}a \Leftrightarrow a\,(R \bullet S)\,c \Leftrightarrow \exists\, b\,(a\,R\,b \wedge b\,S\,c) \Leftrightarrow \exists\, b\,(b\,R^{-1}\,a \wedge c\,S^{-1}b) \Leftrightarrow \exists\, b\,(c\,S^{-1}\,b \wedge b\,R^{-1}\,a) \Leftrightarrow c\,(S^{-1} \bullet R^{-1})\,a$ ■

2.17 – Si $T \subseteq B \otimes C$: $S \subseteq T \Rightarrow R \bullet S \subseteq R \bullet T$ /monotonie

2.18 – $[R \bullet S] = [R]\cdot[S]$ au sens où

$$\forall i,j : [R \bullet S]_{ij} = \bigvee_k (r_{ik} \wedge s_{kj})$$

où $\wedge$ dénote le *et* et $\bigvee_k$ un *ou étendu* à un nombre quelconque d'arguments.

NOTE. Bien distinguer :

- L'*intersection* de relations entre 2 mêmes ensembles, dont la matrice se calcule par produit terme à terme, définit la *conjonction* de 2 relations entre ces mêmes ensembles.
- La *composition*, de relations de A vers B, puis de B vers C, induit une relation de A vers C, correspondant à la *mise en séquence* des 2 relations. La matrice de la composée se calcule par une variante booléenne du produit matriciel classique.

2.5.2 Intérêt

Le produit de composition est un *puissant moyen de définir de nombreuses relations utiles*, calculables à partir d'un nombre réduit de relations génériques qu'il suffira de documenter.

Cette remarque est utilisée aussi bien dans les *réseaux sémantiques* que dans les *bases de données relationnelles et déductives.*

Les informations offertes seront plus nombreuses, bien qu'on en stocke le moins possible, minimisant ainsi les problèmes de cohérence et de mise à jour, dès lors que les informations demandables sont reconstructibles selon des schémas définis à partir des informations stockées.

Ainsi, on minimise les problèmes de maintenance corrective et évolutive, on favorise la qualité et l'étendue de l'information offerte, le coût de la consultation dépendant maintenant du degré d'élaboration de l'information demandée à partir des informations physiquement disponibles.

2.6 EXERCICES

2.6.0

Soient deux ensembles : A, à m éléments, et B, à n éléments. Montrer que $\mathfrak{R}_{AB}$, ensemble des relations de A à B, contient exactement $2^{m \cdot n}$ relations distinctes de A à B. Combien existe-t-il de relations distinctes de B à A ?

2.6.1 Géométrie plane

Soit P l'ensemble des points d'un plan, D l'ensemble des droites du plan, et C l'ensemble des cercles du plan.

Soit R la relation d'appartenance d'un point de P à une droite de D. Soit S la relation de « *contact en un seul point* » entre une droite de D et un cercle de C.

Définir la signification de R•S. L'interpréter en termes de position du point vis-à-vis du cercle.

2.6.2 Gestion d'un parc de voitures

Votre compagnie de taxi doit entretenir un parc de 300 voitures de marque TWK. Une documentation de ce constructeur indique sur quels modèles de voitures parmi les 75 commercialisés sont montés chacun des 17 types de moteur employés, ce qui définit une relation *montéSur* $\subseteq$ Moteurs $\otimes$ Modèles (cf. tab. 2.2 et fig. 2.1).

Soit L la liste des différents modèles de votre parc.

Définir l'ensemble N des moteurs correspondants.

Définir la liste L^+ des modèles de voiture que votre compagnie peut acquérir sans impliquer d'autres modèles de moteur. Montrer que $L \subseteq L^+$.

Après réception d'une petite liste A des moteurs dont la fabrication sera prochainement arrêtée, fournir la nouvelle liste L* des modèles dont l'achat reste possible, et la liste L^- des modèles à revendre d'occasion.

2.7 PROBLÈMES

2.7.0 Plasturgie

On considère une usine de moulage de produits en matière plastique. Définir les relations entre : un ensemble C de commandes, un ensemble O de (types d') objets, un ensemble M de moules (pour un seul type d'objets ou plusieurs), un ensemble P de presses à injecter.

Comment les représenter ? (On supposera $|O|>1000$, $|M|>1500$, $|P|>100$.)

Expliciter des produits de composition entre ces relations. Comment définir quelles machines pourront être utilisées pour une commande particulière ?

(D'après J.F. Petit.)

2.7.1 Communication

On pose que la relation $A \subseteq P \otimes M$ est satisfaite par un couple (p, m) si la personne $p \in P$ a accès au media $m \in M$.

Vérifier que $C = A \bullet A^{-1}$ est une relation interne à P, concernant la possibilité pour deux personnes de communiquer par un media non précisé de M.

Etendre ce raisonnement au cas suivant : soit E un ensemble d'étudiants répartis dans un vaste monde, S l'ensemble des stations de travail auxquels ils ont accès, R l'ensemble des réseaux locaux et autres auxquels ont accès ces stations, P un ensemble de portes entre réseaux. A partir de relations de E à S, de S à R, de R à P et de leurs inverses, quelle(s) relation(s) suffira-t-il de calculer pour savoir si deux étudiants x et y quelconques de E peuvent communiquer via cette *techno jolie* ?

2.8 LECTURES

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien,* tome I, chap. II, Coll. Programmation, Ed. Gauthier-Villars, 1973.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science,* vol. I, chap. II, Coll. Texts and Monographs on Computer Science, Springer Verlag.

GONDRAN M. & MINOUX M., *Graphes et algorithmes,* Collection de la Direction des Etudes et Recherche d'Electricité de France, Eyrolles, Paris, 1985, chap. I.

REINHARDT F. & SOEDER H., *Atlas des mathématiques*, Chap. 3, La Pochothèque, Le Livre de Poche, Paris, 1997 ; traduction de *DTV-Atlas der Mathematik*, Deutscher Taschenbuch Verlag Gmbh, Munich, 1974.

Collectif, *Petite encyclopédie des mathématiques*, Chap. 14, Edition Pagoulatos, Paris, 1980 ; traduction de *Kleine Enzyklopädie der Mathematik*, VEB Bibliographisches Institut, Leipzig, 1975.

CHAPITRE 3

FONCTIONS

La notion de fonction a émergé au XVIIIe siècle, en relation avec la notion de formule ou avec celle de tracé. Ainsi, en 1749, Euler définissait une *fonction* comme une « quantité variable dépendant d'une autre quantité variable ». Au début du XIXe siècle, les travaux de Fourier ont montré l'insuffisance de ce point de vue, puis Cauchy et Dirichlet ont établi une vue plus générale de la fonction comme *correspondance univoque* d'un ensemble à un autre. Par exemple, la réservation obligatoire des places pour un spectacle fait associer à chaque place soit une marque de disponibilité soit un nom de personne ou d'organisme réservataire ; au sens actuel, cette association constitue une fonction de l'ensemble des places vers l'ensemble des marques.

Ce chapitre sera complété par le chapitre 5 sur la calculabilité des fonctions, et le chapitre 6 sur la complexité de ces calculs.

3.1 RELATIONS BINAIRES RÉGULIÈRES

DÉFINITION 3.1 – Soit une relation R. On dira :

R ***régulière à gauche***	*ssi*	chaque point de l'image de R n'a qu'une origine.
R ***régulière à droite***	*ssi*	chaque point du domaine de R n'a qu'un point comme image.
R ***(bi-)régulière***	*ssi*	elle est *régulière à droite et à gauche.*

EXEMPLE. Considérons la relation *est l'arrondi par défaut de* qui, à tout entier relatif z associe chacun des réels r tels que $z \leq r < z+1$. Cette relation est *régulière à gauche* : un réel donné comme 3,742 n'est l'image que d'un entier relatif, ici 3.

PROPOSITION 3.1 – La relation inverse d'une relation régulière à gauche est une relation régulière à droite, et réciproquement.

3.2 FONCTIONS

3.2.1 Présentation

DÉFINITION 3.2 – On appelle ***fonction de A vers B*** toute *relation régulière à droite* de A à B.

Une fonction est une relation telle que chaque point de A a *au plus une image.* Pour de petits ensembles finis, elle peut être définie en extension par une table à 2 colonnes : argument (ou valeur d'origine) / résultat (ou valeur image).

EXEMPLE. Si à chaque étudiant j'associe (au plus) UNE note, la relation *estNoté* est une fonction, souvent définie en extension par une table à 2 colonnes nom/note.

NOTE. On dira souvent d'une telle ***relation régulière à droite*** qu'elle est ***fonctionnelle.***

Fonctions plafond / plancher

On appelle ***plafond*** d'un réel x, et on note $\lceil x \rceil$, l'entier qui lui est égal ou sinon immédiatement supérieur.

On appelle ***plancher*** de x, et on note $\lfloor x \rfloor$, l'entier qui lui est égal ou sinon immédiatement inférieur

Ainsi, pour $x = -7{,}537$: $\lceil x \rceil = -7$ et $\lfloor x \rfloor = -8$.

Ces 2 fonctions sont des fonctions entières de variables réelles, très utiles pour des définitions précises en algorithmie.

EXEMPLE. Soit à daller un espace rectangulaire P·L avec des dalles carrées d·d. Une dalle pouvant être recoupée à dimension, sans que les restes soient réutilisables, il faudra $\lceil P/d \rceil \cdot \lceil L/d \rceil$ dalles, soit pour P = 4m, L = 5m, d = 30 cm :
$\lceil 500/30 \rceil \cdot \lceil 400/30 \rceil = 17 \cdot 14 = 238$ dalles, dont $16 \cdot 13 = 208$ utilisées telles quelles et $17 + 14 - 1 = 30$ à recouper.

3.2.2 Fonctions mathématiques / fonctions en programmation

En programmation, on parle de *fonctions pures* pour les fonctions au sens mathématique, qui *donnent toujours le même résultat pour un argument donné.* C'est le cas des fonctions standard telles que *valeur absolue, sinus, exponentielle, logarithmes divers* ; c'est le cas des fonctions n'utilisant que leurs paramètres et des variables locales, à l'exclusion de toute variable globale ou rémanente.

Dans certains langages de programmation, on appelle à tort fonction toute procédure ayant la même *syntaxe* d'emploi qu'une fonction pure. *Ne sont pas fonctions au sens mathématique les « fonctions » des langages de programmation telles que des appels identiques ne donnent pas toujours le même résultat.* Entrent dans ce cadre des pseudo-fonctions : la plupart des « fonctions » C, les « fonctions » pseudo-aléatoires, d'entrée/sortie, les date, heure ...

Depuis quelques années, on s'oriente vers un nouveau partage des procédures en :

- *fonctions*, nécessairement pures, réconciliant programmation et mathématique, bénéficiant facilement d'optimisations à la compilation ;
- *actions*, procédures ayant le monopole de l'utilisation des variables globales et des effets de bord, assimilables à de nouvelles instructions, traitées moins finement, catégorie dont relève la plupart des pseudo-fonctions.

3.2.3 Applications & fonctions partielles

Relation totale

DÉFINITION 3.3 – Soit une relation R de A à B. On dira :

R ***totale à gauche***	*ssi*	chaque point de A a une image au moins dans B,
R ***totale à droite***	*ssi*	chaque point de B est image d'au moins un point de A.
R ***totale***	*ssi*	elle est totale à droite *et* à gauche.

PROPOSITION 3.2 – La relation inverse d'une relation totale à gauche est une relation totale à droite, et réciproquement.

Applications

DÉFINITION 3.4 – On appelle ***application*** de A vers B toute fonction de A vers B, de domaine A *tout entier.*

Une application est une fonction telle que chaque point de A a *exactement une* image : c'est une relation *totale à gauche.*

On dénotera une application f de A dans B par $f : A \to B$ ou $f \in B^A$.

B^A désigne par convention l'ensemble des applications de A dans B.[1]

EXEMPLE. Si f est une fonction de codage, f est une application dès que *chaque élément de l'ensemble à coder a reçu un code.*

Fonction partielle

DÉFINITION 3.5 – On appelle ***fonction partielle*** une *fonction qui n'est pas une application,* donc fonction *sans image pour au moins un point* de l'ensemble de départ.

EXEMPLE. fact(n) ≡ (si n = 0 alors 1 sinon n·fact(n – 1)) définit la factorielle, qu'on montre (par récurrence) être une application $\mathbb{N} \to \mathbb{N}$. Par contre, dans $\mathbb{Z}$, fact ne définit qu'une fonction partielle : aucun résultat n'est atteint si $n < 0$.

En informatique, une définition partielle sera souvent une définition boguée.

On transforme une fonction partielle f en application

- soit en restreignant l'ensemble de départ au seul domaine de la fonction,
- soit en ajoutant à l'ensemble image, une ou plusieurs valeur(s) symbolique(s) réservée(s) aux cas initialement non définis, comme dans :

  fact(n) ≡ (si n < 0 alors erreurArgumentNégatif sinon
  si n = 0 alors 1 sinon n·fact(n – 1))

C'est le parti adopté dans la norme IEEE 754 concernant l'arithmétique des nombres en virgule flottante : toutes les opérations sont totalement définies ; si un opérande (ou un jeu d'opérandes) viole une condition de calculabilité, le résultat est un code d'erreur non ambigu NaNxxx, formé d'un code préfixe réservé NaN (Not a Number) et d'un suffixe xxx caractérisant l'erreur commise.

Les relations comme applications multivoques

C. Berge présentait en 1958 une relation entre deux ensembles A et B comme application multivoque de A dans B, ou si l'on préfère, *application de A dans l'ensemble des parties de B.* Partant d'un point a de A, celui-ci avait pour image par la relation une partie Γa de B à n(a) points, avec $n(a) \geq 0$: c'est l'origine de la notation Γ introduite au paragraphe 2.1.4. L'absence d'image par une relation R quelconque pour un point x s'interprète alors $\Gamma x = \varnothing$ pour l'application multivoque associée, qui se trouve même dans ce cas parfaitement définie.

3.3 SURJECTION

DÉFINITION 3.6 – Une ***surjection*** ou ***fonction surjective*** est une fonction telle que *chaque point de l'ensemble d'arrivée est l'image d'au moins un point de l'ensemble de départ.* En ce sens, il s'agit d'une *relation totale à droite.*

---

[1] On peut aussi noter $A \to B$ l'ensemble des applications de A dans B ; $f : A \to B$ se lit alors comme $f \in (A \to B)$ ... (G. Boulaye).

On parlera aussi de fonction de A ***sur*** B (par opposition à une fonction de A *dans* B).

EXEMPLE. Si f est une fonction de codage, f est surjective si *tous les codes possibles sont utilisés ; le code est* alors dit *saturé.*

3.4 IMAGE & INVERSE

3.4.1 Notation fonctionnelle

Soit f une fonction définie de A dans B.

Si $x \in A$ et x f y (pour la relation f) alors y est unique et peut être considéré comme résultat d'une transformation de x. C'est pourquoi, selon les contextes, se rajouteront aux notations *prédicatives* classiques f(x,y) ou xfy, des ***notations fonctionnelles*** telles que : y = f(x).

x est alors dit *antécédent* ou *origine* de y ; et y est dit *image de* x par f, ou *valeur* de f *en* x.

3.4.2 Inverses

Soit y = f(x), la notation $x = f^{-1}(y)$ a-t-elle un sens ?

Au sens des RELATIONS, f^{-1} *existe toujours.*

Mais comme f, *en tant que fonction, est une relation régulière à droite, la relation* f^{-1} *est une relation régulière à gauche (prop. 3.1).*

Si f *est une fonction, son inverse* f^{-1} *au sens des relations existe donc sans être nécessairement une fonction.*

Pour que f^{-1} soit aussi une fonction, il faut que f^{-1} soit une relation régulière à droite et donc il faut que f soit une relation régulière à gauche.

3.5 INJECTION & BIJECTION

3.5.1 Injection

DÉFINITION 3.7 – On appelle ***injection*** une *fonction régulière à gauche.*

COROLLAIRE 3.3 – Une ***fonction*** est ***inversible*** (en tant que fonction) *ssi* c'est une injection.

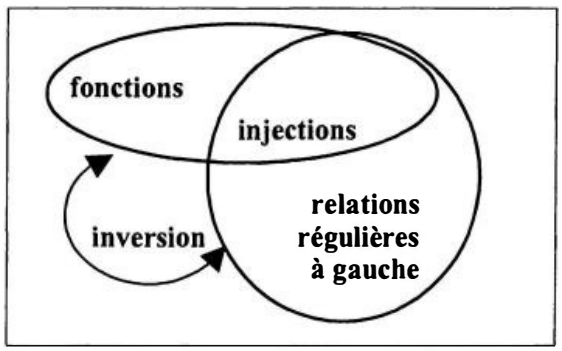


Fig. 3.1 Inversion de fonctions et injections.

PROPOSITION 3.4 – L'inverse d'une injection est une injection.

PREUVE. Par la proposition 3.1, une injection étant une relation régulière à droite ET à gauche, il en est de même de son inverse. ■

3.5.2 Bijection

DÉFINITION 3.8 – Une ***bijection*** ou ***application bijective*** est une *application injective et surjective.*

Une bijection est une relation *totale à gauche* (en tant qu'application) *et à droite* (en tant que surjection).

Elle garantit l'unicité d'origine pour un point image (en tant qu'injection) et l'unicité d'image d'un point origine (en tant qu'application).

Une bijection f entre deux ensembles A et B les met en correspondance exacte et totale, ce qu'on notera souvent f : A↔B.

3.5.3 Propriétés

PROPOSITIONS

3.5 – L'inverse d'une injection f est une fonction ; cette fonction inverse n'est une application (totale à gauche) que si f est surjection (totale à droite). Donc *une fonction inverse n'est une application que si elle est l'inverse d'une injection surjective.*

3.6 – L'inverse d'une injection surjective est une application injective.

3.7 – L'inverse d'une application bijective est une application bijective.

Ces propriétés s'établissent :

- en ramenant la situation donnée aux propriétés de régularité et de totalité,
- en établissant les propriétés correspondantes de l'inverse
- et en retranscrivant ces propriétés en catégories convenables.

EXEMPLE. f injection surjective donne-t-il f^{-1} application injective ?
f est régulière à droite et à gauche (injection), de plus totale à droite (surjection) ; donc son inverse g est régulière à droite et à gauche, et totale à gauche ; donc l'inverse g est injection, et application (application injective).

3.8 – Deux ensembles A et B mis en correspondance par une bijection f ont même cardinalité.

3.5.4 Permutation

DÉFINITION 3.9 – Toute bijection interne à un ensemble fini est dite ***permutation.***

Soit un ensemble E à n éléments, symboliquement $e_1, e_2, \ldots e_n$.

PROPOSITION 3.9 – Il y a n^n applications de E dans E.

En effet, à chaque e_i on peut faire correspondre n e_j différents.

PROPOSITION 3.10 – Il y a n! permutations pour E .

PREUVE. Ces permutations sont des bijections, donc à chaque e_i on doit faire correspondre un e_j *distinct.* A e_1 on peut faire correspondre n e_j différents, à e_2 tout élément de E sauf $p(e_1)$, soit n–1 e_j différents entre eux, à e_3 tout élément de E sauf $p(e_1)$ et $p(e_2)$ ($\neq p(e_1)$) soit n – 2 éléments de E différents entre eux, ... soit finalement $n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot 1 = n!$ permutations possibles. ■

3.6 SYNOPTIQUES

Tableau 3.1 Propriétés des divers types de relations et fonctions.

Nom	Observations	Régularité	Totalité	Schéma
Relation		?	?	
Fonction	Tout point d'origine a au plus une image	Droite	?	
Application	Tout point d'origine a exactement une image	Droite	Gauche	
Surjection	Tout point d'origine a au plus une image, tout point de l'ensemble d'arrivée est image	Droite	Droite	
Injection	Tout point d'origine a au plus une image tout point image a au plus une origine	Droite + Gauche		
Bijection	Tout point d'origine a exactement une image tout point image a exactement une origine	Droite + Gauche	Gauche + Droite	

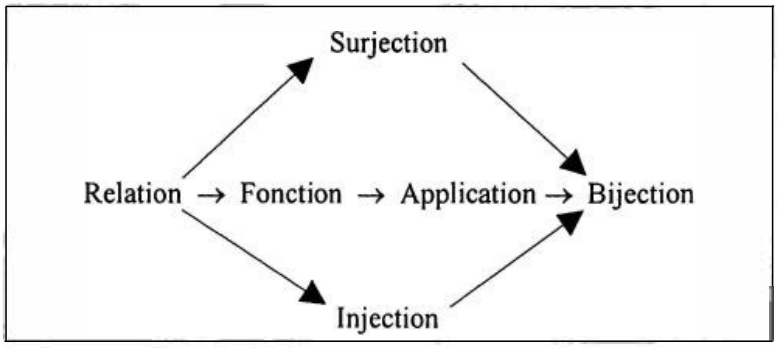


Fig 3.2 Spécialisation progressive, de la relation à la bijection

Tableau 3.2 Inverse des relations & fonctions.

Relation directe			Relation inverse		
Nom	Régularité	Totalité	Nom	Régularité	Totalité
Relation	?	?	Relation	?	?
Fonction	Droite	?	?	Gauche	?
Application	Droite	Gauche	?	Gauche	Droite
Surjection	Droite	Droite	?	Gauche	Gauche
Injection	Droite + Gauche	?	Injection	Droite + Gauche	?
Bijection	Droite + Gauche	Gauche + Droite	Bijection	Droite + Gauche	Gauche + Droite

3.7 COMPOSITION DE FONCTIONS

3.7.1 Présentation

PROPOSITION 3.11 – Soient les fonctions $f : A \to B$ et $g : B \to C$; la relation composée $h = f \bullet g$ est une fonction de A dans C.

PREUVE. Par f tout élément a de A a au plus une image b dans B ; si ce b existe, il a au plus par g une image c dans C : donc chaque a de A a au plus une image c dans C, et $h = f \bullet g$ est une fonction. ■

PROPOSITION 3.12 – Soient les applications $f : A \to B$ et $g : B \to C$; la relation composée $h = f \bullet g$ est une application $h : A \to C$.

PREUVE. Par f tout élément a de A a exactement une image b dans B, qui par g a exactement une image c dans C : donc chaque a de A a exactement une image c dans C, et $h = f \bullet g$ est une application. ■

NOTE. Si $h = f \bullet g$, où h est une fonction, $x\ (f \bullet g)\ y$ se note $y = h(x)$, et signifie $y = g(f(x))$. *Il est classique en théorie des fonctions de noter une telle fonction composée* $h = g \circ f$ (au lieu de $h = f \bullet g$ en théorie des relations).

EXEMPLE. Soit $f : x \to x+1$ et $g : x \to x^2$. Si x vaut successivement 0, 1, 2, 3, 4... f(x) vaut successivement 1, 2, 3, 4, 5... et $h(x) = g(f(x))$ vaut successivement 1, 4, 9, 16, 25... d'où $h = g \circ f = x \to (x+1)^2$.

3.7.2 Propriétés

PROPOSITION 3.13 – Quelles que soient les fonctions $f : A \to B$, $g : B \to C$, $h : C \to D$

$g \circ (f \circ h) = (g \circ f) \circ h$ /associativité de la composition

PROPOSITION 3.14 – L'identité I, qui à tout élément x d'un ensemble A associe le même élément x, est neutre pour la composition.

$$I_A \bullet f = f \bullet I_B = f = f \circ I_A = I_B \circ f$$

3.8 ENSEMBLES DÉNOMBRABLES ET FINIS

3.8.1 Présentation

DÉFINITIONS

3.10 – On dit un ***ensemble dénombrable*** s'il peut être mis en bijection avec $\mathbb{N}$.

3.11 – On dit un ***ensemble fini*** s'il peut être mis en bijection avec un sous-ensemble borné de $\mathbb{N}$; ses éléments se trouvent ainsi numérotés.

EXEMPLES
On prouve que l'ensemble $\mathbb{Z}$ des entiers signés, l'ensemble $\mathbb{N}^2$ des paires d'entiers, l'ensemble des fractions, l'ensemble $\mathbb{Q}$ des rationnels sont *dénombrables* ; il n'en est pas de même de $\mathbb{R}$ ni de $\mathbb{C}$.

$\mathbb{B}$, un ensemble de chiffres, l'ensemble des caractères d'une écriture alphabétique, d'une écriture syllabique sont finis, et donc codables par des entiers bornés.

COROLLAIRE 3.15 – Tout ensemble qui peut être mis en bijection avec le sous-ensemble des entiers de 1 à n est un ***ensemble fini de cardinalité n***.

PREUVE. Par la proposition 3.8. ■

PROPOSITION 3.16 – Soient A, B deux ensembles de cardinalité n : il existe une bijection f pour les mettre en correspondance.

PREUVE. Par la définition 3.11 et le corollaire 3.15, les éléments de A (resp. B) sont numérotables de 1 à n. Ces éléments ayant été numérotés, f est définie en posant :

$$\forall i\ 1 \leq i \leq n \rightarrow f(a_i) = b_i. \ ■$$

3.8.2 Suites indexées, suites finies

Soit D un ensemble quelconque, et I un ensemble dénombrable.

Une application $s : I \rightarrow D$ définit une ***suite d'éléments*** de D ***indexée par*** I ; on dit cette ***suite finie*** si I est fini.

Avec $i \in I$, s(i) sera maintenant noté s_i.

On a alors : $\quad im(s) = \{ s_i \mid i \in I \}$

ou, fréquemment, $\quad im(s) = \{ E(i) \mid i \in I \}$,
où E(i) dénote quelque expression algébrique en i à valeur dans D.

Ainsi notera-t-on $\{ 2n+1 \mid n \in \mathbb{N} \}$ l'ensemble des impairs, au lieu de :

$$\{ s_i \mid i \in \mathbb{N} \} \text{ où } s: \mathbb{N} \rightarrow \mathbb{N} \text{ avec } s(n) = 2n+1\ \forall n \in \mathbb{N}\ . (!).$$

3.8.3 N-uples

N-uples homogènes

DÉFINITION 3.12 – On appelle ***n-uple*** sur D une séquence finie $s = (s_0, s_1, s_2, \ldots s_{n-1})$ telle que pour chaque i de I on ait $s_i \in D$.

N-uples hétérogènes

Soit $F = \{ D_i \mid i \in I \}$ une *suite finie d'ensembles indexée* par I.

DÉFINITION 3.13 – On appelle *n-uple* sur F une suite finie $s = (s_0, s_1, s_2, \ldots s_{n-1})$ telle que pour chaque i de I on ait $s_i \in D_i$.

NOTE. On considérera comme 0-uple l'ensemble vide, 1-uple les singletons, 2-uples les paires ordonnées, 3-uples / 4-uples, les triplets et quadruplets... ; ***tuple*** sera réservé aux n-uples de taille n inconnue.

3.8.4 Produit cartésien généralisé

DÉFINITION 3.14 – On appelle ***produit cartésien généralisé*** des D_i de F l'ensemble $Di_0 \otimes Di_1 \otimes \ldots \otimes Di_{n-1} = \{ s \mid \forall i \in I\ s_i \in D_i\} = \bigotimes_{i \in I} D_i$.

NOTES

- C'est le modèle des articles Pascal (sans variante) ou des structures C (sans union).
- Quand $\forall i \in I$, $D_i = D$, et si $n = |I|$, on pose $D^n = \bigotimes_{i \in I} D_i$.

EXEMPLES

- $\mathbb{B}^m$ est l'ensemble des valeurs que peut prendre un *vecteur booléen* de m bits ; en programmation on assimilera un *mot* de m bits à un tel « vecteur ».
- Soit V un ensemble de notes ; V^m est l'ensemble des valeurs que peut prendre un *vecteur* de m notes. Si chaque note du vecteur a une définition précise liée à son rang, alors les vecteurs assignés à un individu, un système, une solution... pour les représenter, sont dits *profils* de ces entités.

PROPOSITIONS

3.17 – La cardinalité d'un produit cartésien généralisé D est le produit des cardinalités des ensembles composants :

$$D = \bigotimes_{i \in I} D_i \rightarrow |D| = \prod_{i \in I} |D_i|$$

3.18 – Pour un ensemble D, $|D^n| = |D|^n$.

3.9 RELATIONS N-AIRES

3.9.1 Présentation

DÉFINITION 3.15 – Soit $F = (Di_0, Di_1, Di_2, \ldots, Di_{n-1}) = \{ D_i \mid i \in I \}$ une suite finie d'ensembles indexée par I. Tout $R \subseteq Di_0 \otimes Di_1 \otimes \ldots \otimes Di_{n-1}$, *collection de n-uples* de F est dit ***relation n-aire*** sur F, relation dont les D_i sont les ***dimensions***.

EXEMPLES

- On note $m \equiv n \bmod k$ et on dit « m et n sont congrus modulo k » si k divise exactement $|m - n|$: c'est une *relation ternaire* entre entiers.
- En physiologie, psychologie, économie... on utilise une *relation ternaire* dite *de vicariance* : un triplet (X, Y, F) la satisfait *ssi* X supplée Y dans la fonction F.
- En analyse de données, on rencontre une *relation quaternaire* entre objets à classer *(x, y) plus similaires que (z, t)*.

Plus généralement, une relation R n-aire peut être représentée *en extension* par une *table à double entrée*, à raison :

- d'une ligne par n-uple (ou fait élémentaire) satisfaisant R,

- d'une colonne par *dimension de la relation*, ou ensemble D_i membre de F.

|R| est alors donné par le nombre de lignes, tandis que le nombre n de colonnes correspond à l'arité de la relation.

C'est le modèle fondateur des *bases de données relationnelles*.

Tableau 3.3 Définition tabulaire d'une relation quinaire (ou d'arité 5).

Relation CLIENT

– 5 dimensions –

Nom	Adresse	Code Postal	Ville	Numéro-client
***	***	***	***	***
DUPONT	20, Impasse du Progrès	F 69684	Pontcharra sur Turdine	U 017
DURAND	25, Place d'Armes	L 035	Bertrange	Z 043
MARTIN	3, Rue Sans-Souci	F 69425	Millery	W 452
DE WERRA	25, Rue Einstein	CH 0456	Ecublens	Z 018
***				

3.9.2 Projection

DÉFINITION 3.16 – On note $\text{proj}_i(R) \equiv \{ d \in D_i \mid \exists\, r \in R\ r_i = d \}$.

La i-ème ***projection*** d'une relation n-aire, ou projection sur la i-ème dimension, regroupe les éléments de la i-ème colonne de la table.

3.9.3 Décomposition canonique

PROPOSITION 3.19 – Toute relation n-aire R définie sur une suite finie d'ensembles indexée par I se ramène à une suite R_i de n relations binaires indexées par I, où chaque R_i exprime alors la relation entre les n-uples ou lignes de la table s, et leur i-ème colonne ou composante :

$$\forall i \in I\ \forall s \in R\ R_i(s, s_i)$$

Cette proposition permet de transformer une table de relation n-aire en n tables de relations binaires, souvent fonctionnelles, entre les lignes s et leur i-ème attribut : ici

$$\text{Client(nom, adresse, codePostal, ville, numClient)} \equiv$$
$$\exists\ \text{client Nom(client, nom)} \wedge \text{Adresse(client, adresse)} \wedge \text{CodePostal(client, codePostal)}$$
$$\wedge\ \text{Ville(client, ville)} \wedge \text{NuméroClient(client, numClient)}.$$

L'éclatement d'une table de relation n-aire en n tables de relations binaires est notamment utile lorsqu'une dimension est elle-même produit cartésien de sous-dimensions.

3.9.4 Jointure

Soient $F_1 = (A_0, A_1, \ldots A_{k-1} ; B_0, B_1, \ldots B_{m-1})$ et $F_2 = (B_0, B_1, \ldots B_{k-1} ; C_0, C_1, \ldots C_{n-1})$ *deux suites d'ensembles indexées non disjointes.*

A une relation R (k + m)-aire de F_1, et une relation S (m + n)-aire de F_2 on associe une ***jointure*** $R \bowtie S$ définie par :

$$R \bowtie S \equiv \{ (a_0, a_{k-1}, \dots a_{k-1}, b_0, b_{k-1}, \dots b_{k-1}, c_0, c_1, \dots c_{n-1}) \mid (a_0, a_1, \dots a_{k-1}, b_0, b_1, \dots b_{k-1}) \in R \wedge (b_0, b_1, \dots b_{k-1}, c_0, c_1, \dots c_{n-1}) \in S \}$$

REMARQUES

- Si $F_1 = F_2$, la jointure se ramène à l'intersection.
- La jointure est idempotente ; elle peut être considérée comme commutative, et, sous certaines conditions, comme associative (Aho, § VIII.9).
- Chaque relation R peut être considérée comme la jointure de ses relations binaires associées, vues comme partageant un index des n-uples de R.

3.9.5 Schémas de relation et relations

En matière de bases de données, un ***schéma de relation*** S = (I, F) est un couple où I, dit ***ensemble des attributs***, indexe une suite finie F d'ensembles, tout $a \in I$ ayant pour domaine l'ensemble D_a correspondant dans F.

Une *relation n-aire* R apparaît alors comme un couple(S, T) associant

- un schéma S, dont l'ensemble des attributs est de cardinalité n,
- une table T comportant exactement
  - une colonne par attribut de S, de même nom et de même domaine,
  - une ligne par n-uple de R.

3.10 FONCTIONS GÉNÉRALISÉES

3.10.1 Présentation

Tout ce qui a été dit des relations $R \subseteq A \otimes B$ qui, dès lors qu'elles sont régulières à droite, peuvent être vues en tant que fonctions, applications, surjections, injections, bijections, reste vrai si

$$A = D_0 \otimes D_1 \otimes \dots \otimes D_{n-1} \text{ et } B = E_0 \otimes E_1 \otimes \dots \otimes E_{k-1}$$

Une fonction f définie sur A et à valeur dans B a maintenant un domaine dom(f) $\subseteq \otimes D_i$ et une image im(f) $\subseteq \otimes E_j$.

Cette fonction est d'*arité n* si A possède n composantes D_i : elle peut alors être considérée comme ayant soit n arguments distincts, soit un n-uple de A comme argument unique.

EXEMPLES

- Soit D l'ensemble des droites d'un plan ; l'angle entre deux de ces droites peut être considéré comme donné par une fonction *angle* : $D \otimes D \rightarrow \mathbb{R}$.
- Considérons un certain volume rempli d'un fluide, et, à tout instant, en chaque point, les vecteurs vitesse et accélération du fluide en ce point : on peut les considérer globalement comme donnés par une fonction $\mathbb{R}^4 \rightarrow \mathbb{R}^6$.
- Les fonctions généralisées pourront associer au n-uple descriptif d'un objet, d'une solution, le k-uple l'évaluant du point de vue de k critères. Ces profils permettront :

- d'évaluer la *proximité* de profils-type (reconnaissance de formes, identifications et classifications diverses, diagnostics),
- si les critères correspondent à des préférences, des *classements* en vue de choix...

3.10.2 Distances

Ce concept fonde la géométrie, et aussi le concept de similarité.

DÉFINITION 3.17 – La fonction $d : A \otimes A \to \mathbb{R}^+$ est dite ***distance** pour A* ssi :

1. $\forall\ x, y \in A \quad d(x, y) = 0 \leftrightarrow x = y$
2. $\forall\ x, y \in A\ d(x, y) = d(y, x)$ /symétrie
3. $\forall\ x, y, z \in A\ d(x, z) \leq d(x, y) + d(y, z)$ /inégalité du triangle

PROPOSITION 3.20 – Si d est une distance pour A,

$$\forall\ x, y, z \in A \mid d(x, y) - d(y, z) \mid \leq d(x, z)$$

Distance discrète

PROPOSITION 3.21 – Pour tout ensemble A, la fonction $d : A \otimes A \to \mathbb{N}$ définie par *d(x,y) = (si x = y alors 0 sinon 1)* est toujours une distance, dite ***distance discrète.***

NOTE. Cette distance est vraiment grossière, et on utilise le plus souvent en mathématiques discrètes, des distances à valeur entière (§ 3.11.8).

Famille de distances dans $\mathbb{R}^k$

PROPOSITION 3.22 – Pour $A = \mathbb{R}^k$, les fonctions $d_n(x, y) = \sqrt[n]{\sum_{i=1}^{k} |x_i - y_i|^n}$ constituent une famille de distances.

Pour tout couple x, y de points, on a toujours $n \geq m \to d_n(x, y) \leq d_m(x, y)$.

NOTES. Une distance d_n

- est dite ***urbanistique*** ou ***de Manhattan*** si n = 1 : c'est la longueur du plus court chemin formé seulement de segments parallèles aux axes, ce qui évoque les déplacements dans une ville avec plan en grille ;
- est dite ***euclidienne*** ou à ***vol d'oiseau*** si n = 2,
- pour $n = \infty$, est assimilée à $\max_{i=1 \text{ à } k} (|x_i - y_i|)$.

3.10.3 Evaluation partielle

Soit une fonction $f : X \otimes Y \otimes Z \to W$. On peut vouloir fixer le premier argument, et considérer maintenant f(a, y, z) comme une *fonction* particulière *en y et z*, notée $f_{x=a}(y, z) : Y \otimes Z \to W$.

Cette fonction sera dite ***évaluation partielle*** de f pour x = a.

L'opération peut être répétée, et en ce sens la fonction initiale peut être considérée comme une fonction $f : X \to (Y \to (Z \to W))$, menant à $f_{x=a} : Y \to (Z \to W)$, $f_{x=a, y=b} : Z \to W$, puis à la constante $f_{x=a, y=b, w=c}$.

On considère alors successivement x, y... comme des paramètres, les arguments résiduels étant des variables.

EXEMPLE. Soit $f(p, x, y) = x^2 + p \cdot x \cdot y + y^2$.

L'évaluation partielle de f pour $p = -2, -1, 0, 1, 2$ donnera respectivement :

$f_{p=-2}(x, y) = x^2 - 2 \cdot x \cdot y + y^2 = (x - y)^2$,

$f_{p=-1}(x, y) = x^2 - x \cdot y + y^2$,

$f_{p=0}(x, y) = x^2 + y^2$,

$f_{p=1}(x, y) = x^2 + x \cdot y + y^2$,

$f_{p=2}(x, y) = x^2 + 2 \cdot x \cdot y + y^2 = (x + y)^2$

3.11 EXERCICES

3.11.0

Montrer que toute fonction f de A vers B peut s'interpréter :

- comme une application de dom(f) vers B,
- comme surjection de A dans im(f).

3.11.1

Montrer que toute relation de A à B peut s'interpréter :

- comme une application de $A \otimes B$ dans $\{0,1\}$;
- comme une application de A dans 2^B.

3.11.2

Montrer qu'une fonction paire $\mathbb{R} \to \mathbb{R}$ n'est inversible que dans un cas trivial.

3.11.3

Pour un entier signé k, on considère les fonctions $f_k : x \in \mathbb{R} \to x^k \in \mathbb{R}$.

- sont-elles totalement définies ?
- sont-elles inversibles ?

3.11.4

Soit $f : x \to x^2 + 1$; expliciter f o f o f.

3.11.5

Assimiler $\mathbb{N} \otimes \mathbb{N}$ à un quadrillage ; montrer qu'à chaque sommet (p, q) de ce quadrillage on peut associer un numéro n(p,q) distinct. En déduire que $\mathbb{N}^2$ et $\mathbb{Q}$ sont dénombrables.

3.11.6

Soient 2 ensembles : A comporte k éléments, et B en comporte n. Combien y-a-t-il d'applications de A dans B ? combien de fonctions ?

En déduire que $|B^A| = |B|^{|A|}$.

3.11.7

On considère le quadrilatère convexe ABCD, dont les côtés ont pour longueurs respectives AB = 17, BC = 23, CD = 15, DA = 13. Encadrer les longueurs des deux diagonales (i.e. associer à chacune une borne inférieure et une borne supérieure).

3.11.8 Distance de Hamming

Pour sa théorie des codes détecteurs/correcteurs d'erreur, Hamming pose :

$\forall\ x,y \in \mathbb{B}^n\ d_H(x,y) \equiv$ « nombre de composantes par lesquelles x et y diffèrent »

Vérifier que cette fonction d_H est bien une distance entre n-uples booléens.

3.11.9

Soit une table traçante dont la plume peut se déplacer dans deux directions perpendiculaires « en x » et « en y ». Choisir la distance d_i, qui permettra d'estimer au mieux la durée d'un déplacement si les moteurs agissant pour chacune des coordonnées

- doivent travailler successivement,
- peuvent travailler simultanément.

(On négligera les temps de mise en route et d'arrêt des moteurs.)

3.11.10

On considère un plan P muni d'une distance d_i, et un point O du plan.

On appelle «cercles de rayon R, de centre O et d'ordre i» les $C_i(R, O) = \{x \mid d_i(O, x) = R\}$.

Décrire les familles C_1, C_2, C_∞.

Montrer que $0 < i < 1$ donne des familles d'astroïdes à 4 pointes ou losanges curvilignes concaves, $1 < i < 2$ des losanges curvilignes convexes, $i > 2$ des ovales de plus en plus proches, quand i croît, du carré dans lequel ils sont inscrits.

3.12 PROBLÈMES

3.12.1 Devinette numérique

Un camarade vous propose de deviner le nombre auquel vous avez pensé ; il vous dit :

« Pense à un nombre entier de 1 à 30, multiplie-le par l'entier précédent, divise par 2 ; ça fait combien ? »

- vous lui dites 55 ; il répond « c'était 11 » ; il a raison ;
- vous lui dites 300 ; il répond « c'était 25 » ; il a raison ;
- vous lui dites 75 ; il répond « pas possible » ; il a raison.

Expliquer pourquoi.

3.12.2 Emploi du temps

On considère les ensembles

- G de groupes d'étudiants de l'établissement,
- H des heures ouvrables,
- A d'activités,
- S de salles (banalisées ou spécialisées),
- E d'enseignants.

Définir sur ces ensembles la relation d'emploi du temps de la semaine.

En vue d'en extraire divers « emplois du temps » particuliers, présentables sous la forme de tables à 2 dimensions, définir les cas où cette relation se ramène à une fonction de $V \otimes W$ vers $X \otimes Y \otimes Z$.

En prenant par exemple $|G| = 20$, $|H| = 80$, $|A| = 200$, $|S| = 15$, $|E| = 60$, montrer que le nombre d'emplois du temps admissibles est faible devant $|G \otimes H \otimes A \otimes S \otimes E|$ (on suppose qu'un groupe a un emploi du temps de 32 heures par semaine).

Expliciter des contraintes de validité permettant à une relation généralisée $R \subseteq G \otimes H \otimes A \otimes S \otimes E$ d'être un emploi du temps admissible.

3.12.3 Dictionnaires multilingues

On considère le cas de dictionnaires à n langues, comme une relation $L \subseteq A \otimes B \otimes C \otimes D \otimes \ldots$ entre vocabulaires des n langues, définie en extension par m n-

uples ou « équivalences » décrivant un concept suffisamment étroit à raison d'un mot ou locution par langue.

Un premier système consiste à présenter tous ces n-uples dans l'ordre alphabétique de la première langue, puis de la seconde... Montrer que la taille du dictionnaire est alors de l'ordre de $m \cdot n^2$.

Un second système numérote chaque n-uple et est formé :

- de n lexiques d'entrée faisant correspondre à chaque mot ou locution « a » le numéro k de son n-uple ;
- d'un lexique général faisant correspondre à chaque numéro k le n-uple (a, b, c, d, ...) correspondant, explicitant le concept de numéro k dans chacune des langues.

Commenter cette approche en termes de décomposition canonique de L.

En comptant les numéros comme locution dans une (n + 1)-ième langue, montrer que la taille du dictionnaire varie alors en $m \cdot (3n + 1)$.

Pourquoi cette formule est-elle la plus répandue à partir de quatre langues?

Les considérations précédentes restent-elles valides si on considère que chaque entrée participe en moyenne à μ (>1) n-uples ?

3.12.4 Géométries

Soit P un demi-plan limité par une droite D, et soit C l'ensemble des demi-cercles de P centrés sur D. On considère C comme représentant l'ensemble des *alignements* dans une géométrie particulière, alignements définis par 2 points distincts de P–D.

On notera a(x,y,z) le fait que x, y, z soient alignés, z faisant partie de l'alignement défini par x et y.

Etablir une condition *cond* telle que : $a(x, y, z) \wedge a(x, z, t) \wedge cond \rightarrow a(x, y, t)$.

Donner quelques autres propriétés de a.

Donner une définition des relations *coupe* et *parallèle à* de ces alignements, qui justifient le postulat de cette géométrie : « par un point hors d'un alignement A passe un alignement parallèle à A et une infinité d'alignements non sécants avec A ».

(Cf. modèle euclidien de Poincaré pour la géométrie de Lobatchewski, in Pabion, *Logique mathématique*, Hermann, op.cit.)

3.12.5 Induction fonctionnelle

Vous soupçonnez un phénomène p de dépendre des variables x et y. Vous faites 2 expériences en fixant une des valeurs.

Vous trouvez $p_{x=4} = 7y - 3$; et $p_{y=2} = 4x - 5$.

Ces expériences sont-elles compatibles avec l'existence d'une fonction p(x, y) ?

Peut-on admettre l'existence d'une fonction $p(x, y) = q + r \cdot x + s \cdot y$?

Peut-on admettre l'existence d'une fonction de la forme $p(x, y) = f(x) \cdot g(y)$?

Supposons qu'en réalité $p(x, y)= q + r \cdot x + s \cdot y + t \cdot xy$. Montrer que les deux « expériences » ne suffisent pas à la reconstitution de $p(x, y)= q + r \cdot x + s \cdot y + t \cdot xy$.

Quel est alors le statut des solutions antérieures ?

3.13 LECTURES

AHO A., ULLMAN J., *Concepts fondamentaux de l'informatique*, chap. VII, section 7 et suivantes, chap. VIII, Dunod, Paris, 1993.

BERGE C., *Principes de combinatoire*, Dunod, 1968.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science*, vol. I, chap. II, Coll. Texts and Monographs on Computer Science, Springer Verlag.

LIVERCY, *Théorie des programmes,* Dunod, Paris, 1978.

REINHARDT F.& SOEDER H., *Atlas des mathématiques*, chap. 3, La Pochothèque, Le Livre de Poche, Paris, 1997 ; traduction de *DTV-Atlas der Mathematik*, Deutscher Taschenbuch Verlag Gmbh, Munich, 1974.

Collectif, *Petite encyclopédie des mathématiques*, chap. 14 et 5, Edition Pagoulatos, Paris, 1980 ; traduction de *Kleine Enzyklopädie der Mathematik*, VEB Bibliographisches Institut, Leipzig, 1975.

CHAPITRE 4

RELATIONS BINAIRES INTERNES

4.1 PRÉSENTATION

Un ensemble A se présente comme une collection d'éléments. Lorsque A modélise un ensemble concret, on peut avoir à rendre compte d'une correspondance existant, d'un certain point de vue, entre ses éléments. Cette correspondance est précisément une *relation,* interne à A. Souvent, on considérera que la relation *structure* l'ensemble, au sens du point de vue adopté.

DÉFINITION 4.1 – Une relation binaire R sera dite ***relation binaire interne*** à A *ssi* $R \subseteq A \otimes A$.

EXEMPLES

- P, ensemble de Personnes, pereOuMere = {(p1, p2) ∈ P⊗P | p1 pèreOuMèreDe p2}, relation interne à P.
- Relation de divisibilité, notée a | b pour « a divise b », sur l'ensemble des entiers non nuls.

4.1.1 Représentations

Matrice caractéristique

La ***table booléenne*** ou ***matrice caractéristique*** d'une relation binaire interne R est maintenant une matrice booléenne *carrée* [R] qui comporte, en supposant les éléments de A numérotés :

- une i-ème ligne par élément $a_i \in A$ considéré comme origine,
- une j-ième colonne par élément $a_j \in A$ considéré comme image,
- pour chaque couple (i, j) un élément

Tableau 4.1 Matrice caractéristique de la relation de divisibilité dans {1, ... , 8}.

divise	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	o	1	o	1	o	1	o	1
3	o	o	1	o	o	1	o	o
4	o	o	o	1	o	o	o	1
5	o	o	o	o	1	o	o	o
6	o	o	o	o	o	1	o	o
7	o	o	o	o	o	o	1	o
8	o	o	o	o	o	o	o	1

$$[R]_{ij} = r_{ij} = (\textit{si } (ai, aj) \in R \textit{ alors } 1 \textit{ sinon } 0)$$

Graphes

Pour les diagrammes sagittaux ou graphes d'une relation interne (étudiés en détail dans la partie Graphes), on confond ensemble de départ et d'arrivée. Le graphe visualise alors la structuration de l'ensemble support par la relation sous étude.

Le graphe G = <A, R> associé à la relation R de l'ensemble A

- considère A comme ensemble de ***points*** ou ***sommets*** du graphe,
- considère R comme l'ensemble des flèches ou ***arcs*** de G, (appelés ***boucles*** s'ils relient un point à lui-même : fig. 4.1).

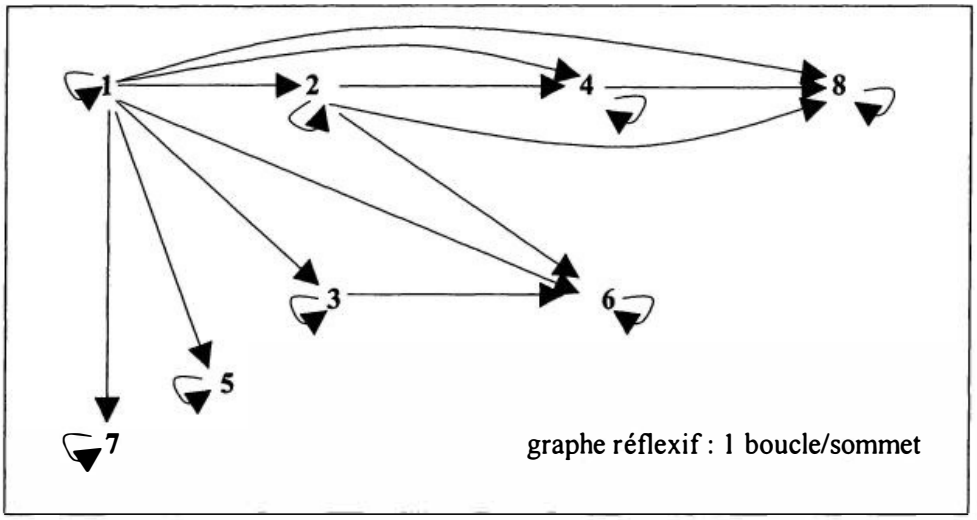


Fig. 4.1 Graphe de divisibilité des entiers de 1 à 8.

4.1.2 Relations remarquables

Tableau 4.2 Relations remarquables pour un ensemble A.

Nom	Notée	Définition	Matrice	Graphe
relation nulle ou vide	0_A	jamais satisfaite	0 partout	points isolés, sans boucle
identité	I_A	satisfaite par chaque élément avec lui-même	matrice unité (0 partout, sauf diagonale à 1)	points isolés avec 1 boucle par point
relation universelle	U_A	toujours satisfaite	1 partout	graphe complet, i.e. comprenant la totalité des arcs possibles

4.2 PROPRIÉTÉS DES RELATIONS BINAIRES INTERNES

On n'examinera que les plus urgentes. Chacune d'elles se répercute de façon spécifique sur son graphe, qui recevra le même qualificatif.

Ainsi, graphe réflexif/ transitif/ symétrique/... signifiera graphe d'une relation réflexive/ transitive/ symétrique/...

4.2.1 Réflexivité

DÉFINITION 4.2 – Pour une relation R interne à A, on pose :

R ***relation réflexive*** $\equiv \forall\, x \in A\ xRx$
R ***relation irréflexive*** $\equiv \forall\, x \in A\ \neg\, xRx$
R ***relation non réflexive*** $\equiv (\exists x \in A\ xRx) \wedge (\exists y \in A\ \neg yRy)$

PROPOSITIONS

4.1 – Toute relation *réflexive* a sa matrice caractéristique de diagonale 1.

EXEMPLES. Relations I_A, U_A ; analogies, préordres larges, équivalences, ordres larges...

4.2 – Tout graphe réflexif comporte une boucle par sommet (fig. 4.1).

4.3 – Toute relation *irréflexive* a sa matrice caractéristique de diagonale 0.

EXEMPLES. Relation 0_A ; perpendicularité entre droites d'un plan, antériorités entre tâches d'un projet ; (pré)ordres stricts...

4.4 – Les graphes *irréflexifs* ne comportent que des sommets sans boucles.

4.5 – La diagonale de la matrice caractéristique d'une relation *non réflexive* comporte au moins un 1 et au moins un 0.

Réflexivité et modélisation

La qualité de *réflexivité / irréflexivité* est souvent arbitraire, et fixée par *convention explicite* en fonction de la seule commodité d'exploitation.

Au contraire, la *non-réflexivité* sera une *irrégularité informative* à préserver.

4.2.2 Symétrie

DÉFINITION 4.3 – Pour une relation R interne à un ensemble A, on pose :

R ***relation symétrique*** $\equiv \forall\, x,y \in A\ xRy \rightarrow yRx$
R ***relation asymétrique*** $\equiv \forall\, x,y \in A\ xRy \rightarrow \neg yRx$
R ***relation antisymétrique*** $\equiv \forall\, x,y \in A\ xRy \wedge yRx \rightarrow x=y$
R ***relation non symétrique*** sinon.

EXEMPLES

- *Symétries :* relations 0_A, I_A, U_A ; perpendicularité entre droites ; analogies, équivalences.
- *Antisymétries :* ordres larges.
- *Asymétries :* filiations, (pré)ordres stricts, précédences entre tâches.

PROPOSITION 4.6 – Toute relation *symétrique* a sa matrice caractéristique symétrique par rapport à sa diagonale principale.

NOTE. Aux relations symétriques on associera un ***graphe symétrique***, obtenu en remplaçant chaque paire d'arcs opposés par un trait démuni de flèche, appelé ***arête***.

PROPOSITION 4.7 – Toute relation *asymétrique* est *irréflexive*.

PREUVE. Soit R une relation asymétrique, et « a » un point tel que aRa ; par l'asymétrie de R, on aurait $aRa \rightarrow \neg aRa$, et aRa serait contradictoire ; il n'existe donc aucun point pour lequel on ait aRa. Donc R est irréflexive. ■

4.2.3 Transitivité

DÉFINITION 4.4 – On dit d'une relation R interne à un ensemble A :

R ***relation transitive*** $\equiv \forall\, x,y,z \in A\ xRy \wedge yRz \rightarrow xRz$
R ***relation intransitive*** $\equiv \forall\, x,y,z \in A\ xRy \wedge yRz \rightarrow \neg\, xRz$
R ***relation non transitive*** sinon.

EXEMPLES

- les relations 0_A, I_A, U_A sont *transitives,* de mêmes que les préordres, ordres et équivalences ;
- la relation entre droites $\perp$ (*être perpendiculaire*) est *intransitive* ;
- la relation $\neq$ est *non transitive.*

REMARQUE. Soit [R] une matrice booléenne caractéristique d'une relation interne R. Considérons dans cette matrice le rectangle ayant pour sommets les cases (j, j), (i, j), (j, k), (i, k) :

- r(j, j) est sur la diagonale,
- r(i, j) est sur la même colonne que r(j, j),
- r(j, k) est sur la même ligne que r(j, j),
- finalement, le sommet diagonalement opposé à r(j, j) désigne r(i, k).

La relation représentée est

- *transitive* si r_{jk} est à 1 *chaque fois* que r_{ij} et r_{jk} sont à 1 ; ex : $r_{24}=1 \wedge r_{48}=1 \rightarrow r_{28}=1$
- *intransitive* si r_{jk} est à 0 *chaque fois* que r_{ij} et r_{jk} sont à 1 ;
- *non transitive* sinon.

Tableau 4.3 Matrice d'une relation transitive.

				j↓				k↓
divise	1	2	3	4	5	6	7	8
1	*1*	1	1	1	1	1	1	1
i→2	0	*1*	0	1	0	1	0	1
3	0	0	*1*	0	0	1	0	0
j→ 4	0	0	0	*1*	0	0	0	1
5	0	0	0	0	*1*	0	0	0
6	0	0	0	0	0	*1*	0	0
7	0	0	0	0	0	0	*1*	0
8	0	0	0	0	0	0	0	*1*

NOTE. Le test de transitivité est accéléré en transcrivant la définition

$$\forall x \forall y\ xRy \rightarrow (\forall z\ yRz \rightarrow xRz)$$

PROPOSITION 4.8 – *Toute relation intransitive est irréflexive.*

PREUVE. cf. § 0.2.6, ex. 11. ■

EXEMPLE. La *perpendicularité* entre droites est une relation *intransitive donc irréflexive.*

Modélisation

La transitivité s'interprète souvent en termes de *cohérence* ou de conséquence naturelle.

REMARQUE. Les graphes transitifs sont très « chevelus ». Pour y voir clair, on dessine souvent un graphe simplifié de la relation, où ne figurent pas les arcs déductibles par transitivité ou réflexivité, en faisant par ailleurs mention évidente de la transitivité (resp.

réflexivité). Strictement, ce graphe simplifié s'obtient en substituant à la relation transitive R une relation plus fine (ou relation squelette), définie par $R' = R - I$ et $R_S = R' - R'^2$. En retour, $R = R^*_S$ si R est réflexive, et $R = R^{\oplus}_S$ si R est irréflexive (chap. 9).

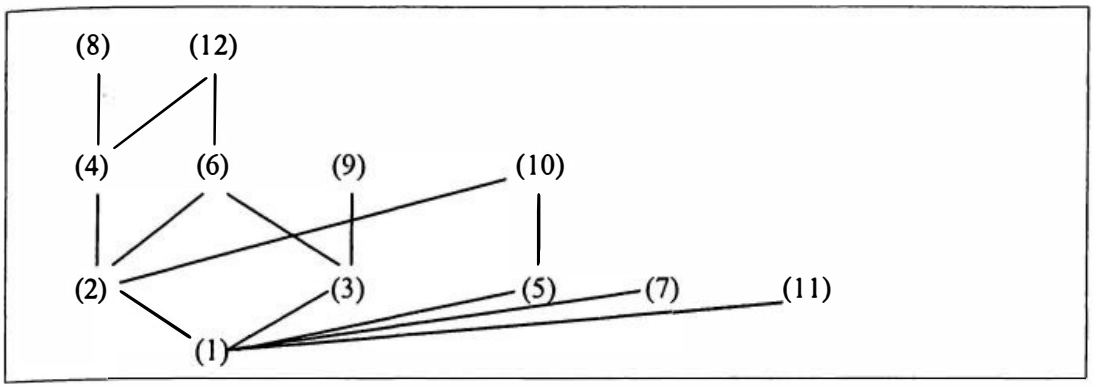


Fig. 4.2 Squelette d'une relation réflexive et transitive (divisibilité de 1 à 12).

4.2.4 Retour sur la finesse

Union, *intersection*, *inclusion*, *finesse*, *composition* de relations s'adaptent facilement au cas où ensemble de départ et d'arrivée sont identiques.

Cependant, les propriétés nouvellement définies et la relation d'identité introduisent de nouveaux résultats.

Propriétés de la finesse entre relations internes

PROPOSITIONS

4.9 – $0_A \subseteq I_A \subseteq U_A$.

4.10 – R est *réflexive* ssi $I_A \subseteq R$.

4.11 – R est *symétrique* ssi $R = R^{-1}$.

DÉFINITION 4.5 – On appelle ***symétrisée*** de R : $R^S \equiv R \cup R^{-1}$.

PROPOSITIONS

4.12 – Si R est la symétrisée de R, $R \subseteq R^S$ et $R^{-1} \subseteq R^S$.

4.13 – R^S est la relation symétrique la plus fine majorant R (et R^{-1}).

4.14 – $I_A \bullet R = R = R \bullet I_A = R$.

4.15 – R est *transitive* ssi $R \bullet R \subseteq R$.

PREUVE. Par les définitions de la composition (2.8) et de la transitivité (4.4). ■

4.16 – Soient deux relations $R, T \subseteq A \otimes A$

$$0_A \subseteq R \cap T \subseteq R \subseteq R \cup T \subseteq U_A$$
$$0_A \subseteq R \cap T \subseteq T \subseteq R \cup T \subseteq U_A$$

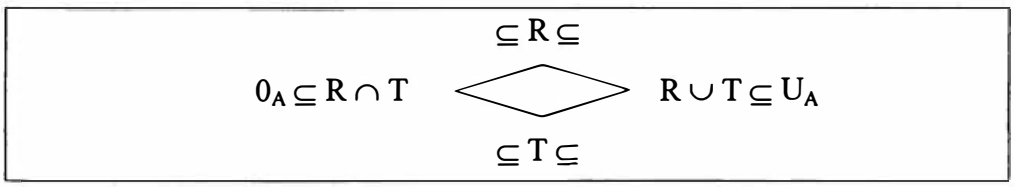


Fig. 4.3 Diagramme en losange pour la finesse entre relations, leurs intersection et leur union.

4.3 RESTRICTION

DÉFINITION 4.6 – Soit une relation $R \subseteq A \otimes A$, et B un sous-ensemble de A. On pose :

$$R_{|B} \textbf{\textit{ restriction}} \text{ de R à B } \equiv R_{|B} = R \cap (B \otimes B)$$

PROPOSITION 4.17 – Entre la restriction de R à un ensemble E, et R on a toujours

$$R_{|E} \subseteq R$$

EXEMPLES

- une relation définie sur $\mathbb{R}$ peut être restreinte à $\mathbb{N}$;
- une relation de jumelage entre villes peut être restreinte à un sous-ensemble de ces villes.

4.4 ÉQUIVALENCE

4.4.1 Présentation

DÉFINITION 4.7 – On appelle ***équivalence*** toute *relation binaire interne réflexive symétrique et transitive.*

EXEMPLES

- dans tout ensemble A, l'*identité* I_A, la *relation universelle* U_A ;
- la *congruence modulo k* dans $\mathbb{N}$, k étant considéré comme paramètre ;
- le *parallélisme* entre droites du plan, noté a // b, en admettant que toute droite est parallèle à elle-même ;
- la *similitude entre triangles* du plan, en admettant que tout triangle est semblable à lui-même.

4.4.2 Postulat d'Euclide

PROPOSITION 4.18 – *2 éléments équivalents à un même troisième sont équivalents entre eux.*

PREUVE. Soit $\approx$ une équivalence dans A.
Pour tous les a, b, c de A nous avons :

$a \approx c \wedge \underline{b \approx c} \rightarrow a \approx c \wedge \underline{c \approx b}$ par symétrie de $\approx$

$a \approx c \wedge c \approx b \rightarrow a \approx b$ par transitivité de $\approx$

soit finalement : $a \approx c \wedge b \approx c \rightarrow a \approx b$. ■

4.4.3 Représentations

La *matrice* carrée représentative *d'une équivalence est diagonale par blocs.*

Elle ne comporte que des 0, *sauf des blocs carrés ne comportant que des 1,* centrés sur la diagonale principale, sous l'éventuelle réserve d'une permutation des lignes et des colonnes reflétant un réarrangement des éléments de l'ensemble.

Le ***graphe*** associé est formé de ***sous-graphes disjoints***. Chacun de ces sous-graphes est une ***clique*** (sect. 7.7).

Blocs et cliques correspondent aux classes d'équivalence définies au paragraphe 4.4.5.

Tableau 4.4 Matrice d'une équivalence (blocs en gris).

1	1	o	o	o
1	1	o	o	o
o	o	1	1	1
o	o	1	1	1
o	o	1	1	1

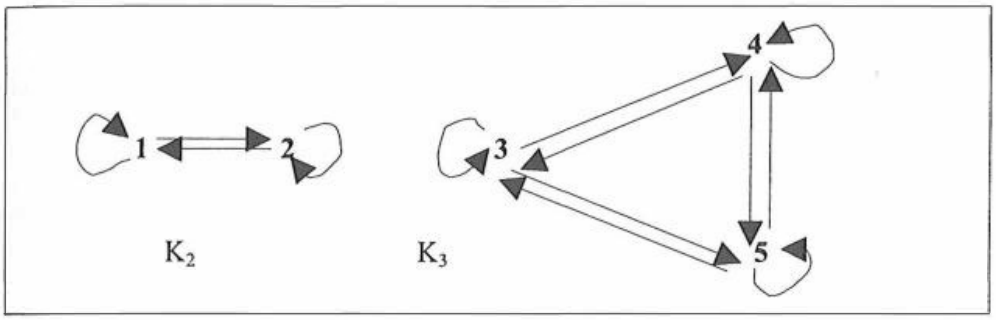


Fig. 4.4 Cliques de l'équivalence du tableau 4.4.

4.4.4 Partition attachée à une équivalence

PROPOSITION 4.19 – A toute équivalence de E, ensemble fini, est attachée une partition Π de E.

PREUVE. Soit un ensemble E muni d'une équivalence $\approx$.

Soient a, b deux éléments de E, et soit $C_a = \{ x \in E \mid a \approx x \}$ et $C_b = \{ x \in E \mid b \approx x \}$ les sous-ensembles de E regroupant les éléments de E équivalents à a, respectivement à b.
C_a et C_b sont non vides puisqu'ils contiennent au moins a (resp. b) et que $\approx$ est réflexive.
Montrons que C_a et C_b sont disjoints ou confondus.
S'ils ne sont pas disjoints, ils ont au moins un élément en commun, disons c. Alors a et b sont équivalents (prop. 4-17). Par la même proposition, tout $x \in C_b$ étant équivalent à b, l'est à a, et appartient à C_a ; donc $C_b \subseteq C_a$. Symétriquement, tout $x \in C_a$ étant équivalent à a, l'est à b, et donc appartient à C_b ; donc $C_a \subseteq C_b$.
Donc, C_a et C_b sont confondus si un c commun existe ; sinon C_a et C_b sont disjoints par définition.
Ainsi, tout a appartient à un et un seul C_a.
Par suite, $C = \{ C_x \in P(E) \mid x \in E \wedge C_x = \{ y \in E \mid x \approx y \} \}$ est une partition de E.

On peut la construire par épuisement :

- On choisit un élément a de E.
- On bâtit $C_a = \{ x \in E \mid a \approx x \}$. Si C_a ne recouvre pas tout E, on choisit $b \in E - C_a$, puis on bâtit $C_b = \{ x \in E \mid b \approx x \} = \{ x \in E - C_a \mid b \approx x \}$. De même, si E n'est pas épuisé, on choisira $c \in E - C_a - C_b \ldots$

Ce procédé est fini dès que E est fini, car le nombre de classes est au pire égal au nombre d'éléments (pour l'identité). Choisir à une certaine étape un autre élément ne pourra que changer l'ordre d'énumération des C_x, mais pas leur définition. ■

4.4.5 Ensemble-quotient

DÉFINITIONS

4.8 – Soit E un ensemble muni d'une équivalence $\approx$. On note **E/≈** la partition associée, dite ***quotient de E par l'équivalence ≈.***

4.9 – Tout ensemble de E/≈ est dit ***classe d'équivalence*** dans E pour $\approx$.

4.10 – Dans chacune des classes C d'un ensemble-quotient E/≈, on peut choisir un élément r_c qui sera dit ***représentant*** de C. Le choix d'un représentant pour une classe est en principe arbitraire.

EXEMPLES

- Pour les entiers naturels, la *congruence modulo k* mène à une partition de $\mathbb{N}$ en k classes C_i ayant pour représentant $i \in \{ 0, 1, 2, \dots , k-1 \}$, l'un des k restes distincts possibles.
- Pour l'ensemble des droites d'un plan, la relation d'équivalence // (*est parallèle à)* mène à une partition des droites, à raison d'*une classe par direction* du plan.

4.4.6 Equivalence attachée à une partition

PROPOSITION 4.20 – *A toute partition Π de E est attachée une équivalence.*

PREUVE. Soient x, y deux éléments de E.
Posons $x \approx y$ ssi il existe un ensemble $C \in \Pi$ telle que $x \in C$ et $y \in C$.
La relation $\approx$ est *réflexive et symétrique.*
Supposons maintenant que pour $x, y, z \in E$ on ait $x \approx y$ et $y \approx z$. Alors il existe un ensemble C_1 de Π tel que $x \in C_1$ et $y \in C_1$ et un ensemble C_2 de Π tel que $y \in C_2$ et $z \in C_2$.
Π étant une partition, y appartient à un ensemble de Π et un seul : donc $C_1 = C_2$, x et z appartiennent au même ensemble, et $x \approx z$. Ce qui établit la *transitivité.*

$\approx$ est donc bien une équivalence induite par Π sur E. ■

4.4.7 Equivalence et partition attachées à une application

DÉFINITION 4.11 – Soit $f : A \to B$ une application, et posons :

$$\forall\ x,y \in A\ (x \approx_f y) \equiv (\ f(x) = f(y)\ )$$

La relation $\approx_f$, dite ***isovalence par f,*** est généralement notée ***ker(f).***

PROPOSITION 4.21 – Une *isovalence* ker(f) est une *équivalence.*

PREUVE. Notons $\approx$ l'isovalence ker(f) ; il vient :
$\forall\ x \in A,\ f(x) = f(x) \to x \approx x$ /réflexivité de = → réflexivité de ≈
$\forall\ x, y \in A,\ (f(x)=f(y) \to f(y) = f(x)) \to (x \approx y \to\ y \approx x)$ /symétrie de = → symétrie de ≈
$\forall\ x, y, z \in A,\ x \approx y \wedge y \approx z \to x \approx z$ /transitivité de = → transitivité de ≈

≈, c'est à dire ker(f), *est donc équivalence dans A du fait que = en est une dans B.* ■

EXEMPLES

- Si $f : A \to \mathbb{N}$, f *numérote* les éléments de A, et A est partitionné en classes C_n regroupant les éléments de A de même numéro n.
- Si f est une fonction de *codage,* l'équivalence ker(f) groupe en une même classe (d'ambiguïté) les éléments de même code.

- Si f est une fonction d'*indexation*, l'équivalence ker(f) définit les classes d'éléments de même clé.
- Si f est une fonction de *notation* liée à un critère, l'équivalence ker(f) définit les classes d'ex aequo.

PROPOSITION 4.22 – A toute application surjective $f : A \to B$, on peut associer une application bijective $f^* : A/\ker(f) \to B$.

PREUVE. Soit r(C) le représentant d'une classe C dans A/ker(f) ; par définition de l'isovalence, $f(r(C)) = f(a)$ pour tout élément a de la classe C. On peut donc bâtir une fonction $f^* : A/\ker(f) \to B$, en posant $f^*(C) = f(r(C))$, pour chaque classe C de A/ker(f), supposé fini.

La fonction $f^* : A/\ker(f) \to B$ obtenue est

- injective, car chaque classe C distincte a une image $f^*(C) = f(r(C))$ distincte par définition,
- surjective comme f,

donc finalement elle est bijective.■

4.5 ORDRES

4.5.1 Ordre large

DÉFINITION 4.12 – On appelle ***ordre large*** toute *relation binaire interne réflexive antisymétrique et transitive.*

EXEMPLES

- Dans l'ensemble des entiers, la relation *divise,* notée | est un ordre large.

  PREUVE. De sa définition $\forall a, b \in \mathbb{N}\ a \mid b \leftrightarrow (\exists p \in \mathbb{N}\ a.p = b)$ on tire
  $\forall a \in \mathbb{N}\ a \mid a$ car $\forall a\ a.1 = a$ /réflexivité de |
  $\forall a, b \in \mathbb{N}\ (a \mid b \wedge b \mid a) \to a = b$ /antisymétrie de |
  $\forall a, b, c \in \mathbb{N}\ (a \mid b \wedge b \mid c) \to a \mid c$ /transitivité de |. ■

- Sont ordres larges *l'inclusion entre sous-ensembles* d'un ensemble, *la finesse entre relations, entre couvertures, entre partitions.*
- $\leq$ dans $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$.

4.5.2 Fonctions monotones

DÉFINITION 4.13 – Soit $f: A \to B$ une fonction d'un ensemble A, ordonné par $\angle$, vers B ordonné par $\preccurlyeq$. On dira :

f ***isotone*** $\equiv$ $\forall x,y \in A\ x \angle y \to f(x) \preccurlyeq f(y)$
f ***antitone*** $\equiv$ $\forall x,y \in A\ x \angle y \to f(y) \preccurlyeq f(x)$
f ***monotone*** $\equiv$ f isotone OU f antitone

4.5.3 Ordre total / partiel

DÉFINITIONS

4.14 – On dira ***ordre total*** sur A un ordre large sur A $\ll$ tel que :

$$\forall x, y \in A\ x \ll y \vee y \ll x$$

Sinon, l'ordre sera dit ***ordre partiel***.

4.15 – Un ensemble acceptant un ordre total est souvent appelé ***chaîne*** ou ***échelle.***

NOTE. Un tel ensemble sera souvent représenté sur un axe orienté.

EXEMPLES

- dans $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ la relation $\leq$ est un *ordre total* ;
- dans l'ensemble des entiers, la relation divise, notée |, est un *ordre partiel,* car 3 ne divise pas 7, et 7 ne divise pas 3 ;
- l'inclusion entre sous-ensembles d'un ensemble, la finesse entre relations, entre couvertures, entre partitions sont respectivement des *ordres partiels.*

REMARQUE. De nombreux problèmes pratiques se ramènent à la recherche d'un *ordre total* $\ll$ *compatible avec un ordre partiel* $\preccurlyeq$, i.e tel que $\forall$ x, y $x \preccurlyeq y \rightarrow x \ll y$.

4.5.4 Ordre strict

DÉFINITION 4.16 – On appelle ***ordre strict*** toute *relation binaire interne asymétrique et transitive.*

PROPOSITIONS

4.23 – *Tout ordre strict est irréflexif* (du fait de l'asymétrie).

4.24 – *Un ordre strict est un ordre partiel,* car il rejette la comparaison d'un objet avec lui-même.

4.25 – *A tout ordre large* $\leq$ *correspond un ordre strict* $<$ obtenu en posant :

$$\begin{aligned}(a < b) &\equiv \forall a,b \quad ((a \leq b) \wedge \neg(b \leq a)) \\ &\equiv \forall a, b \quad ((a \leq b) \wedge (a \neq b))\end{aligned}$$

4.26 – A tout ordre large total correspond un ***ordre*** strict $<$ dit ***linéaire***, i.e. tel que :

$$\forall\, a, b \;\; a < b \vee a = b \vee b < a$$

NOTE. Cette version affaiblie de la totalité conserve la notion d'échelle, et la possibilité de représentation selon un axe orienté.

EXEMPLES. Dans $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ la relation $<$ est l'ordre *strict* linéaire associé à $\leq$. Mais l'inclusion stricte d'ensembles est un ordre strict (non linéaire).

4.5.5 Ordres induits sur des tuples

On considère des tuples de $A \otimes B \otimes \ldots K$ basés sur des *ensembles ordonnés* $<A, \leq_A>$, $<B, \leq_B>$ … $<K, \leq_K>$. Ces ordres sur les dimensions induisent-ils un ou des ordres sur les tuples ?

Ordre unanime

DÉFINITION 4.17 – Une ***relation d'infériorité unanime*** *(resp. supériorité / préférence)* $\preccurlyeq$ entre tuples est satisfaite par deux tuples quand *chaque paire de composantes de même dimension respecte l'ordre total ou partiel de la dimension correspondante.*

On pose par exemple dans $A \otimes B \otimes C$:

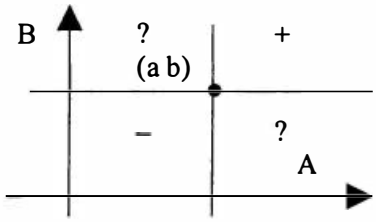
$$(a\ b\ c) \preccurlyeq (f\ g\ h) \equiv a \leq_A f \wedge b \leq_B g \wedge c \leq_C h$$

PROPOSITION 4.27 – L'*infériorité (resp. supériorité / préférence) unanime* est un *ordre partiel entre n-uples.*

REMARQUES

L'extension unanime de $\leq$ à $\mathbb{C}$ ou $\mathbb{R}^n$ n'est toujours qu'un ordre partiel.

Si chaque dimension est munie d'un ordre total, à partir d'un n-uple donné l'espace des n-uples comprend 2^n angles solides (ou ***orthants***) différents, dont 1 de n-uples « supérieurs », 1 de n-uples « inférieurs », et $2^n - 2$ de n-uples « non comparables ».



Quadrants

+ : couples supérieurs	à (a b)
– : couples inférieurs	à (a b)
? : couples non comparables	à (a b)

Fig. 4.5 Ordre unanime à 2 composantes :
quadrants de comparabilité associés à un couple (a, b).

Ordre lexicographique

DÉFINITION 4.18 – Entre tuples de A⊗B⊗...K basés sur des *ensembles ordonnés* $<A, \leq_A>$, $<B, \leq_B>$... $<K, \leq_K>$, on appellera ***ordre lexicographique*** une relation qui

- ordonne les tuples sur leur première composante,
- sinon, en cas d'égalité, sur la seconde,
- sinon, en cas d'égalité, sur la suivante,
- et ainsi de suite jusqu'à la dernière.

EXEMPLE. Posons :

$$(a\ b\ c\ d) \preccurlyeq (x\ y\ z\ t) \equiv a <_A x \vee (a = x \wedge (b <_B y \vee (b = y \wedge (c <_C z \vee (c = z \wedge d \leq_D t)))))$$

Si A = B = ... = K, (a b c d) et (x y z t) se comparent comme deux entrées abcd et xyzt d'un dictionnaire.

PROPOSITION 4.28 – *L'ordre lexicographique est un ordre.* Cet ordre est total *ssi* l'ordre de chaque dimension est total.

4.5.6 Minoration/majoration

Minorant/majorant

DÉFINITION 4.19 – Soit E un ensemble muni d'un ordre $\angle$ et F un sous-ensemble de E.

On appelle ***minorant*** de F tout élément m de E tel que $\forall x \in F\ m \angle x$.
On appelle ***majorant*** de F tout élément M de E tel que $\forall x \in F\ x \angle M$.

Elément minimal/maximal

DÉFINITION 4.20 – Un élément est ***minimal*** (resp. : ***maximal***) dans un ensemble ordonné s'il n'admet pas d'élément minorant (resp. : majorant) dans cet ensemble.

Minimum / maximum

DÉFINITION 4.21 – Un élément est ***minimum*** (resp. : ***maximum***) d'un ensemble ordonné s'il est *le seul élément minimal (resp. : maximal) de l'ensemble.*

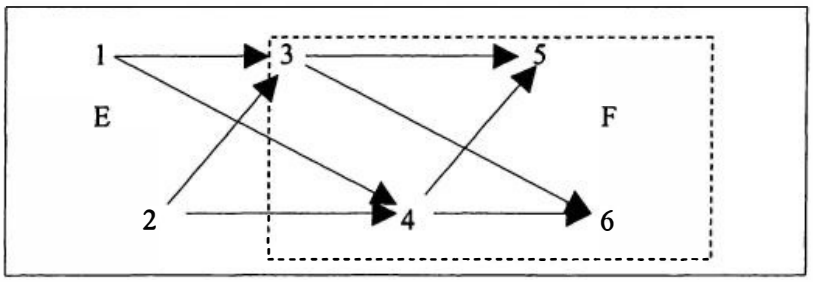


Fig. 4.6 Eléments minorants et minimaux

EXEMPLES

- Dans la figure 4.6, 1 et 2 sont minorants de F ; 3 et 4 sont minimaux dans F ; F ne possède pas de minimum.
- $\mathbb{N}_2 = \{2, 3, 4, 5, \ldots\}$, ordonné par la relation *divise*, admet comme éléments minimaux les nombres premiers. Dans $\mathbb{N}$, $\mathbb{N}_2$ admet 1 comme minorant.
- $\mathbb{N}_1 = \{1\} \cup \mathbb{N}_2$ ordonné par la relation divise, admet 1 comme minimum.
- L'ensemble des relations sur un ensemble E, ordonné par la finesse, admet 0_E comme minimum, U_E comme maximum.
- L'ensemble des relations réflexives (resp. des préordres, des ordres larges, des équivalences) sur un ensemble E, ordonné par la finesse, admet I_E comme minimum, U_E comme maximum, 0_E comme minorant.

4.5.7 Demi-treillis & treillis

DÉFINITION 4.22 – Soit E un ensemble muni d'un ordre $\angle$. Si ***pour chaque paire x, y*** d'éléments de E

- *il existe un plus grand minorant*, dit borne inférieure, et noté inf(x, y), alors $\angle$ structure E en ***inf-demi-treillis,***
- *il existe un plus petit majorant*, dit borne supérieure, et noté sup(x, y), alors $\angle$ structure E en ***sup-demi-treillis,***
- *il existe un plus grand minorant* **et** *un plus petit majorant,* alors $\angle$ structure E en ***treillis.***

EXEMPLES

- $\mathbb{N}_1 = \{1, 2, \ldots\}$, ordonné par la relation divise, a une structure de treillis, avec pour chaque paire (j, k) :

  $$\text{pgcd}(j,k) \mid j,\ \text{pgcd}(j,k) \mid k, \text{ et } j \mid \text{ppcm}(j,k),\ k \mid \text{ppcm}(j,k)$$

  pgcd et ppcm donnent ainsi les bornes *inf* et *sup* cherchées.
- L'ensemble des parties d'un ensemble E, ordonné par l'inclusion, forme un treillis où $\cup$ et $\cap$ associent à toute paire de sous-ensembles les bornes cherchées.
- L'ensemble des relations sur un ensemble E, ordonné par la finesse, forme un treillis où $\cup$ et $\cap$ associent à toute paire de relations les relations-bornes cherchées.

- Par contre, la figure 4.6 ne contient pas de treillis : car les paires {1, 2}, {3, 4} ont des paires {3, 4}, {5, 6} de majorants immédiats non comparables, et elles n'ont donc pas de borne supérieure ; de même pour les minorants.

PROPOSITION 4.29 – Un ensemble muni d'un ordre total est doté d'une structure de treillis.

REMARQUES

- $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ sont des treillis pour les relations $\leq$ et $\geq$, max et min fournissant les bornes.
- La propriété de treillis sera en fait le plus souvent recherchée comme *propriété forte de certains ordres partiels.*
- *En particulier,* $\mathbb{N}^n$, $\mathbb{Z}^n$, $\mathbb{Q}^n$, $\mathbb{R}^n$ sont des treillis non triviaux pour les ordres unanimes étendant aux n-uples les relations = et $\geq$. Les bornes inf et sup sont fournies par les max et min calculés composante à composante.
- Un treillis issu d'un ordre partiel associe à chaque paire d'éléments (x, y) une maille ou losange lui associant (inf(x, y), sup(x, y)) (fig. 4.7).

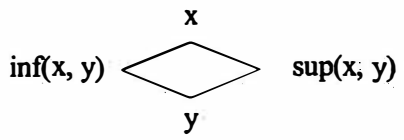


Fig. 4.7 Maille d'un treillis : motif en losange.

4.6 PRÉORDRE

4.6.1 Présentation

Les préordres sont des relations de définition moins contraignante que celle d'ordre ou d'équivalence : on les rencontre donc plus souvent, avec un risque de propriétés trop faibles.

L'intérêt pratique des préordres vient de ce que *chacun engendre une équivalence*, puis un ordre sur l'ensemble-quotient associé, ce qui permet d'interpréter les préordres comme des *ordres sur des paquets.*

D'autre part, un procédé standard défini au chapitre 9 (fermetures transitives) permettra d'associer à toute relation, si quelconque soit-elle, le préordre le plus fin la majorant.

L'*analyse de données* exploite ces deux remarques pour donner aux relations empiriques quelconques le sens le plus clair possible moyennant la plus faible distorsion.

A de telles relations, souvent impossibles à interpréter directement, on associera donc le préordre majorant le plus fin, puis une équivalence induisant une partition de l'ensemble initial, enfin un ordre entre ses classes, exprimant la relation d'origine en termes de rapports (i) entre paquets (ii) à l'intérieur de ces paquets.

Ce procédé permettra une lecture rationnelle aussi exacte que possible de relations expérimentales n'ayant a priori ni rime ni raison.

DÉFINITION 4.23 – On appelle ***préordre*** toute *relation binaire interne réflexive et transitive.*

REMARQUES

- Tout ordre est un préordre.
- Toute équivalence est un préordre.
- Comme pour les ordres, on parlera de préordre *large* (réflexif) ou *strict* (irréflexif) ; et de préordre *total* ou *partiel.*

4.6.2 Préordre induit par une fonction

PROPOSITION 4.30 – Soit $f : A \rightarrow B$ une application, et soit $\leq$ un ordre défini sur B. Si on pose : $\forall\ x,y \in A\ x \angle y \equiv f(x) \leq f(y)$, alors $\angle$ *est préordre pour A.*

PREUVE

- pour tout x de A, on a $x \angle x$ car $f(x) = f(x)$ / réflexivité de $\angle$
- pour tout x, y, z de A, on a $(x \angle y \wedge y \angle z) \rightarrow x \angle z$ du fait de la transitivité de $\leq$ / transitivité de $\angle$
- pour tout x, y, z de A, on a $x \angle y \wedge y \angle x \rightarrow f(x) = f(y)$ du fait de l'antisymétrie de $\leq$; mais on ne peut en déduire en général que $x = y$ car *2 objets ou individus ne sont pas identiques du seul fait qu'ils ont une note, ou un aspect commun.*

Il n'y a donc pas en général antisymétrie de $\angle$, et *c'est en cela que le préordre et l'ordre diffèrent.* ■

4.6.3 Préordres liés aux notations et profils

Soit $f : A \rightarrow N$ une application de A dans N, et soit $\leq$ un ordre total défini sur N, ensemble des notes.

Les considérations précédentes s'appliquent, et les éléments de A se trouvent préordonnés par une telle notation.

Soit maintenant une *liste de critères* dont chacun introduit sa propre notation en termes d'intérêt : les membres de A reçoivent ainsi un tuple de notes, ou ***profil.***

Celui-ci se trouve muni d'un ordre unanime dès lors que chaque critère fournit un ordre de notation, et les profils étant ordonnés en tant que tuples, un *préordre « de domination »*[1] est induit entre membres de A, par :

$$x \rhd y \quad \equiv \quad \text{profil}(x) \geq \text{profil}(y)$$

qui se lira : x domine y ssi le profil de x est supérieur au profil de y.

4.6.4 Domination et décision multicritère

Une décision multicritère s'articule ainsi en 2 étapes : l'*aide à la décision* réduit l'ensemble des solutions à l'ensemble des solutions dominantes, et la *prise de décision* sélectionne parmi ces dernières la solution finalement retenue.

Aide à la décision

Cette phase bâtit à partir des ordres relatifs aux divers critères un *ordre unanime partiel d'intérêt* entre les profils établis, qui induit un *préordre partiel de domination* ou *surclassement* entre les solutions à comparer. Celles-ci se trouve réparties en

[1] Ou « de surclassement ».

- ***solutions dominantes***, i.e. dominées par aucune autre,
- ***solutions dominées***, qui le sont par au moins une autre.

PROPOSITION 4.31 – Soient s et s′ des solutions dominantes ; si s′ est meilleure que s pour un ou plusieurs critères, alors elle est pire que s pour un ou plusieurs autres critères (par l'absurde).

Dans l'espace des critères, l'ensemble des solutions est assimilable à un nuage de points. On clarifie la situation sans perdre de solution intéressante en substituant à l'ensemble des solutions une *liste réduite* formée des seules solutions dominantes :

- chacune des solutions abandonnées est inférieure au sens de chacun des critères retenus à au moins une des solutions conservées ;
- ce n'est vrai d'aucune des solutions restantes qui, par la proposition 4.31, sont ou indifférentes ou non comparables.

L'ensemble de ces solutions dominantes constitue un ***ensemble optimal au sens de Vilfredo Paretto.***

Prise de décision

Plus subjective, la *prise de décision* veut dégager parmi les solutions dominantes UN choix final. L'introduction d'un *arbitrage entre critères* transformera le préordre partiel en un *préordre total*, reflétant les compromis acceptés.

Pour cela, les divers critères sont agrégés par une fonction évaluant l'intérêt global d'une solution, tous critères combinés[2]. Cette fonction définit dans l'espace des critères :

- un préordre total,
- des *isovales* ou courbes (resp. surfaces) d'isovalence, correspondant à autant de *niveaux d'intérêt*,
- les solutions d'intérêt maximal (resp. de coût minimal ou de moindre mal).

Exemple

Soit à préparer un voyage. Diverses solutions sont décrites dans le tableau 4.5 en termes de durée et de coût, grandeurs qui varient en sens inverse de leur intérêt : on cherche les faibles coûts et les courtes durées.

1/ Analyse

Indépendamment de tout arbitrage temps/argent, *il suffit de retenir (a, c, e)* (fig. 4.8, tab. 4.6) :

- b est éliminé par c, plus rapide ET moins chère ;
- de même, d est éliminée par e.

Tableau 4.5 Les solutions à comparer.

Solution	Coût	Durée
a	10 000	5h
b	9 000	7h
c	8 000	6h
d	7 000	15h
e	6 000	12h

[2] On lui demandera seulement d'être monotone (resp. croissante) par rapport à chacune de ses variables, représentant autant de critères.

2/ Décision

La décision est ensuite prise sur la base de l'optimisation d'une *fonction d'intérêt* (resp. de coût), qui définit la note attribuée à chaque solution restante en vue du classement final.

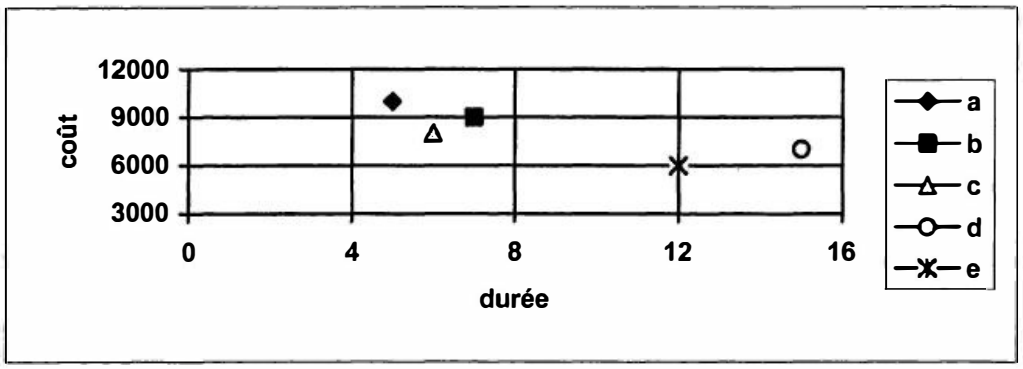


Fig. 4.8 Les solutions dans l'espace des critères.

Tableau 4.6 Solutions dominantes / dominées.

Solution	Coût	Durée	Domination
a	10 000	5h	*dominante*
b	9 000	7h	dominée par c
c	8 000	6h	*dominante*
d	7 000	15h	dominée par e
e	6 000	12h	*dominante*

Diverses politiques sont résumées dans le tableau 4.7 .

Une *première politique* minimise un *coût généralisé* coût + $·durée, où $ représente la *contre-partie d'une heure perdue*, la valeur de $, toujours positive, variant selon qu'il s'agit du calcul d'une société pour ses cadres, ou du temps perdu en début ou fin de vacances. Les courbes isovales d'arbitrage, à intérêt constant, seraient ici formées d'un réseau de *droites* parallèles.

La solution a sera optimale si le temps est précieux, e si la durée a peu d'importance, c dans une large plage intermédiaire de valeurs pour $, voire de styles de décision.

Une *seconde politique* ne tolère coût ou durée que si l'autre facteur est très faible : elle cherche à optimiser le *produit* coût·durée. Les isovales d'arbitrage à intérêt constant seraient ici formées d'un réseau d'*hyperboles équilatères*, et donneraient c comme premier choix, a comme second.

Une *politique* NINI, *refusant* coût et durée, pourrait

- normaliser coût et durée par rapport à leurs médianes (ici 8.000 et 7),
- minimiser (coût normalisé)^2 + (durée normalisée)^2,

et correspondrait à des courbes isovales formant un réseau d'*ellipses* de centre 0. Ici, c serait sur l'ellipse la plus centrale, et se présenterait comme le *meilleur refus* de la durée et du coût.

Tableau 4.7 Recherche d'un *moindre mal* selon diverses politiques.

Solution	Coût	Durée	Choix minimisant coût + $·durée	Choix minimisant Coût·Durée	Choix Nini
a	10 000	5h	Si $ > 2 000F/h	50 000	2,0725
c	8 000	6h	Pour $ de 333F/h à 2 000 F/h	*48 000*	*1,73*
e	6 000	12h	Si $ < 333F/h	72 000	3,66

L'intérêt variant en raison inverse du coût et de la durée, il n'y a pas ici lieu d'envisager le célèbre « rapport qualité/prix ».

4.6.5 Equivalence associée à un préordre

PROPOSITION 4.32 – Soit un *préordre* $R \subseteq A \otimes A$: la relation $\approx$ définie sur A par

$$\approx \equiv R \cap R^{-1} \textit{ est une équivalence}$$

PREUVE. $\approx$ est nécessairement symétrique du fait de sa définition ; et réflexive et transitive car R l'est. ■

EXEMPLES

- Dans un préordre de notation, il s'agira de l'équivalence entre *ex aequo*.
- Dans un préordre général de domination, il s'agit de l'équivalence entre *solutions indifférentes* au regard des critères.

4.6.6 Ordre induit sur l'ensemble-quotient

L'équivalence $\approx$ associe à A un ensemble-quotient $A/\approx$.
Dans cet ensemble-quotient, introduisons une relation $\preccurlyeq$ définie par :

$$\forall\, C_x, C_y \in A/\approx\ (\, C_x \preccurlyeq C_y \leftrightarrow x\, R\, y\,)$$

où C_x, C_y désignent les classes d'équivalence pour $\approx$ attachées à x et y dans A.

PROPOSITION 4.33 – La relation $\preccurlyeq$ est un ordre large.

PREUVE. La relation est *transitive et réflexive* comme R. Elle est *antisymétrique* car $(C_x \preccurlyeq C_y) \wedge (C_y \preccurlyeq C_x) \Rightarrow xRy \wedge yRx \Rightarrow x \approx y \Rightarrow C_x = C_y$

PROPOSITION 4.34 – *Tout préordre sur un ensemble A induit*

(a) une équivalence $\approx$ dans A,

(b) un ordre large sur l'ensemble-quotient $A/\approx$ relatif à cette équivalence.

C'est pourquoi un *préordre* est souvent lu comme un *ordre entre des paquets*, ces paquets étant les classes attachées à l'équivalence induite.

EXEMPLE. On soumet divers produits plus ou moins comparables à un groupe d'utilisateurs, qui notent chaque produit de A à E sur 3 critères essentiels.

Ainsi chaque produit reçoit un profil, par exemple BAC, signifiant « note B (resp. A, C) pour le critère 1 (resp 2, 3) », et il existe maintenant entre ces profils un ordre partiel qui s'induit des ordres de notation.

Posons $x \approx y \leftrightarrow$ profil(x) = profil(y). $\approx$ est une équivalence entre produits de même profil, et chaque profil définit une classe d'équivalence : tous les produits de profil BAC sont perçus de façon similaire par le groupe d'utilisateurs, ils forment une classe de profil BAC.

Ces classes sont ordonnées entre elles comme les profils correspondants, BAC $\leq$ BAB entraîne $C_{BAC} \leq C_{BAB}$, et l'on dira la classe de produits BAC inférieure à la classe BAB. Les classes dominantes seront maximales pour cet ordre.

Sauf gros effectif et répartition uniforme, il n'y aura pas forcément des produits dans les 125 classes possibles. Les classes absentes mettront en contraste les classes présentes et faciliteront leur qualification, menant pour chacune à des recommandations d'emploi différentes.

4.7 RÉCAPITULATIFS

Le tableau 4.8 présente, regroupées par aspects, la liste des propriétés des relations, et de leurs définitions.

Tableau 4.8 Principales propriétés des relations binaires internes.

	Propriété	Définition
Réflexivité	réflexive	$\forall s\ sRs$
	irréflexive	$\forall s\ \neg sRs$
	non réflexive	$(\exists s\ sRs) \wedge (\exists t\ \neg tRt)$
Symétrie	symétrique	$\forall s\ sRt \rightarrow tRs$
	asymétrique	$\forall s\ sRt \rightarrow \neg tRs$
	antisymétrique	$\forall s\ sRt \wedge tRs \rightarrow s = t$
	non symétrique	$(\exists s,t\ sRt \wedge tRs \wedge s \neq t) \wedge (\exists s,t\ sRt \wedge \neg tRs \wedge s \neq t)$
Transitivité	transitive	$\forall s,t,u\ sRt \wedge tRu \rightarrow sRu$
	intransitive	$\forall s,t,u\ sRt \wedge tRu \rightarrow \neg sRu$
	non transitive	$(\exists s,t,u\ sRt \wedge tRu \wedge sRu) \wedge (\exists s,t,u\ sRt \wedge tRu \wedge \neg sRu)$

Le tableau 4.9 rassemble ci-après les grandes catégories de relations, qui jouissent de deux ou plusieurs de ces propriétés.

Tableau 4.9 Propriétés des principales catégories de relations.

Catégories de relations		Aspects des relations		
		Réflexivité	Symétrie	Transitivité
Équivalence		réflexive	symétrique	transitive
Ordre	large	réflexive	antisymétrique	transitive
	strict	irréflexive	asymétrique	transitive
Préordre	large	réflexive		transitive
	strict	irréflexive		transitive
Analogie		réflexive	symétrique	

4.8 EXERCICES

4.8.0

Montrer que la relation nulle est irréflexive, symétrique, asymétrique et antisymétrique, transitive et intransitive, ordre strict.

4.8.1

Caractériser en tant que relation la relation *être perpendiculaire à* entre droites d'un plan.

4.8.2

Soit un ensemble A muni d'une structure de treillis par l'ordre large total $\leq$. Montrer que l'ensemble A^n des n-uples se trouve alors doté d'un ordre *partiel* lui donnant une structure de treillis, dès lors que l'on introduit:

$$\sup((a_1, a_2, \dots a_n),(b_1, b_2, \dots b_n)) = (\max(a_1, b_1), \max(a_2, b_2), \dots , \max(a_n, b_n))$$

et une définition similaire en *min* pour la borne *inf*érieure.

4.8.3

Montrer qu'à tout ordre large est associé un ordre strict ; montrer qu'on peut de même associer à tout préordre, considéré comme large, un *préordre strict.*

4.8.4

Soit un ensemble E muni d'une distance d. On se donne une constante r, et on définit une relation de proximité π par :

$$\forall x, y \in E \; x\pi y \equiv d(x, y) \leq r$$

Montrer que cette relation est réflexive et symétrique (analogie).

Montrer que l'inégalité du triangle $\forall x, y, z \in E \; d(x,z) \leq d(x,y) + d(y,z)$ ne suffit pas à garantir la transitivité de π.

Montrer que π sera cependant une équivalence *ssi*

$$\forall x, y, z \in E \; d(x,z) \leq \max(d(x,y), d(y,z)) \quad \text{(d est alors dite } \textit{ultramétrique}\text{)}$$

4.8.5

On introduit une notion de *circularité* de relation avec les définitions suivantes :

R *circulaire* : $\forall a,b,c\ aRb \wedge bRc \rightarrow cRa$
R *anti-circulaire* : $\forall a,b,c\ aRb \wedge bRc \rightarrow \neg cRa$

Transcrire les grandes définitions de ce chapitre en termes de réflexivité, *circularité*, symétrie.

4.8.6

En vue de construire un logiciel d'arbitrage pour le jeu *Le Compte est Bon*, on envisage d'engendrer automatiquement les formules f constructibles à l'aide d'une liste d'entiers L et des 4 opérateurs arithmétiques usuels, en vue de retrouver le nombre recherché N, ou de s'en rapprocher le plus possible.

On appelle *défaut* d'une formule f la quantité

$$\text{défaut}(f) = |\ \text{valeur}(f) - N\ |$$

et, pour deux formules f et g, on pose $f \preccurlyeq g \equiv \text{défaut}(f) \leq \text{défaut}(g)$.

Montrer que $\preccurlyeq$ est un préordre. Comment s'en servir pour trouver la ou les meilleures formules au fur et à mesure de leur génération?

4.8.7

Soit un ensemble fini X muni d'une relation R. Quelle doit être la nature de R pour qu'elle justifie un *tri* de X ? Justifier le tri de couples (clé, information).

4.8.8

Soit un ensemble D de droites. Caractériser au sens de ce chapitre les relations (être perpendiculaire à) et // (être parallèle à). Sont-ce des ordres? des équivalences ? des préordres ?

4.8.9

Soit un ensemble T de tâches à effectuer dans un projet. Et soit $\prec$ la relation de précédence entre ces tâches ; montrer que $\prec$ est transitive, asymétrique, irréflexive.

4.8.10

Expliquer comment l'ordre lexicographique (signe, exposant, mantisse) permet de comparer directement les nombres virgule flottante normalisés sans addition ni soustraction.

4.8.11

On considère un ensemble P de personnes, muni de la relation « aime ».

Discuter de la transitivité de cette relation .

- sous une hypothèse de fraternité universelle,
- sous une hypothèse de jalousie universelle,
- dans un tiers-cas.

Etendre éventuellement à la réflexivité, à la symétrie.

4.8.12

A Ivanbourg, le Camarade Premier Secrétaire, du fait de..., est nécessairement le plus intelligent de tous les habitants. D'autre part, du fait de..., le Camarade Premier Secrétaire est nécessairement le moins intelligent de tous les habitants. De ce fait, tous les habitants sont identiques : c'est pourquoi ils s'appellent tous Ivan Ivanov... [commenter].

4.8.13 Analogie

On appelle parfois analogie une relation réflexive symétrique.

Soit un ensemble E muni d'une distance d. On se donne une constante r, et on définit une relation de proximité π par : $\forall x, y \in E\ x\pi y \equiv d(x,y) \leq r$.

Montrer que cette relation est une *analogie.*

Montrer qu'une relation qui est analogie et préordre est une équivalence.

4.8.14 Dénombrements

On considère un ensemble X à 5 objets.

Combien admet-il de relations distinctes bâties sur lui ? de relations réflexives ? irréflexives ? de relations symétriques ? asymétriques ? antisymétriques ? d'analogies ?

4.8.15

Vérifier que $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ sont des treillis pour les relations $\leq$ et $\geq$, max et min fournissant les bornes.

En déduire que $\mathbb{N}^n$, $\mathbb{Z}^n$, $\mathbb{Q}^n$, $\mathbb{R}^n$ sont des treillis pour les relations $\leq$ et $\geq$ étendues aux n-uples, max et min calculés composante à composante fournissant les bornes.

Généraliser en montrant que l'ordre $\angle$ obtenus sur les n-uples de $A\otimes B\otimes C\ldots$ à partir des ordres $\angle_x$ définis sur les A, B, C... dote $A\otimes B\otimes C\ldots$ d'une structure de treillis dès lors que chaque couple $(X, \angle_x)$ forme déjà un treillis.

4.8.16

Considérons l'ensemble $F = \{f \mid f : \mathbb{R}\rightarrow\mathbb{R}\}$ des fonctions réelles à valeur réelle.

On pose $f\approx g$ (ou $f = O(g)$) dans F si $\lim_{x\rightarrow\infty} (f(x)/g(x)) = k \neq 0$.

Montrer qu'il s'agit d'une relation d'équivalence (dite « varie comme ») dans F.

Montrer que pour cette équivalence tout polynôme à une variable x, de degré n admet comme représentant x^n.

On pose $f \langle g$ (ou $f = o(g)$) dans F si $\lim_{x\rightarrow\infty} (f(x)/g(x)) = 0$.

Montrer que $\langle$, dite « varie moins vite que », est un ordre strict dans F.

Montrer que pour cet ordre tout polynôme à une variable de degré n varie moins vite que tout polynôme de degré strictement supérieur à n.

4.8.17

Caractériser la relation $\approx$ définie par :

$$a \approx b \Leftrightarrow \exists x\ (a \neq x)\wedge(x \neq b)$$

selon qu'il y a 2, 3, 4... éléments dans l'ensemble considéré.

4.8.18

Soient A, B deux ensembles, et R une relation de A à B.

Montrer que $S = (R\bullet R^{-1})$ est une relation interne à A (resp. que $U = (R^{-1}\bullet R)$ est interne à B).

Montrer que S est symétrique.

Montrer que $dom(S) = dom(R)$, et que $x \in dom(R) \leftrightarrow xSx$.

4.9 PROBLÈMES

4.9.1 (d'après I.C. Lerman)

Soit un ensemble E muni d'une *distance* d qualifiée d'*ultramétrique*, c'est-à-dire satisfaisant une inégalité du triangle renforcée :

$$\forall x, y, z \in E\ d(x,y) \leq \max(d(x,z), d(z,y))$$

On se donne une constante r, et on s'intéresse aux boules $B_{x,r}$ de centre x et de rayon r :

$$\forall\, y \in E \;\; y \in B_{x,r} \leftrightarrow d(x,y) \leq r$$

Montrer que, à r donné, ces boules constituent une partition de E.

Montrer que lorsque r croît depuis 0, elles constituent une chaîne de partitions de moins en moins fine, allant de I_E à U_E.

Soit maintenant une liste d'objets dont on demande une classification rationnelle. Leurs analogies et différences sont résumées par la table de distances 4.10.

Peut-on dire que cette distance est *ultramétrique* ?

En déduire une classification arborescente des objets considérés.

Tableau 4.10 Distances.

d	a	b	c	d	e
a	0	15	9	9	5
b	15	0	15	15	15
c	9	15	0	7	9
d	9	15	7	0	9
e	5	15	9	9	0

4.9.2

La société Forex est chargée d'enquêter sur les préférences d'utilisateurs pour une liste de produits.

Dans un échantillon E de 50 utilisateurs, chaque personne interrogée donne un ordre de préférence pour 5 produits au plus. On dresse ensuite une matrice des préférences P, *chaque p_{ij} indiquant le nombre exact de sondés ayant classé i avant j* (tab. 4.11).

Tableau 4.11 Préférences.

P	a	b	c	d	e	f
a	0	15	14	9	18	20
b	7	0	6	13	9	6
c	8	9	0	11	15	11
d	7	10	8	0	9	5
e	5	15	11	9	0	9
f	3	17	8	3	15	0

On définit une famille de relations P_k par :

$$x\, P_k\, y \text{ ssi } k \text{ personnes au moins préfèrent } x \text{ à } y$$

Etudier les propriétés de ces relations quand le seuil k varie de N à 1. Montrer que ces relations constituent une chaîne de finesse.

Montrer que l'on a en général une propriété de *sous-transitivité* du style :

$$xP_py \wedge yP_qz \rightarrow xP_sz$$

où s dépend de p, q et N. Exprimer cette dépendance par une fonction.

Tout reposant sur le tableau initial, déduire de cette étude des *tests de vraisemblance* permettant de contrôler sa validité.

4.9.3

Vous êtes chargé de rédiger les statuts d'une assemblée à pouvoir réglementaire, dont on souhaite des décisions cohérentes au sens suivant.

Si l'assemblée déclare A prioritaire sur B, et par ailleurs B sur C, on souhaite qu'elle déclare si possible A prioritaire sur C, *et qu'à tout le moins elle ne déclare pas C prioritaire sur A.*

On suppose chaque membre votant cohérent, en ce sens que s'il a voté pour A prioritaire sur B, et pour B prioritaire sur C, alors il vote pour A prioritaire sur C.

Montrer que :

- une majorité simple ou des 2/3 ne permet pas d'atteindre ces objectifs (cf. Condorcet) ;

- une majorité supérieure (5/7 ou 3/4) interdit les circularités mais pas les lacunes ;
- seule l'unanimité garantit la transitivité des préférences.

Quel est l'impact pratique, sachant que l'exigence de majorités élevées est une cause d'immobilisme?

4.9.4

On définit la congruence modulo k par :

$$x \equiv y \bmod k \leftrightarrow k \mid \mathrm{abs}(x - y)$$

En déduire que $m \mid n \rightarrow (x \equiv y \bmod n \rightarrow x \equiv y \bmod m)$, et qu'il existe ainsi un *ordre de finesse* entre congruences.

En donnant le statut convenable à l'égalité et à la relation universelle, assimilée à une congruence modulo 1, on établira que les congruences modulo k forment un treillis calqué sur le treillis de divisibilité.

On en déduira

- qu'une congruence modulo n peut se ramener à des congruences plus simples quand n n'est pas premier,
- que toute équation entière peut se ramener à des équations de congruence modulo k.

4.9.5

On appelle entiers *quadrat-frei* les entiers n'admettant pas de facteurs au carré autres que 1.

Vérifier que 6, 15, 30, 210, 30 030 sont de tels nombres, mais ni 25 ni 180.

Vérifier que l'ensemble des diviseurs d'un quadrat-frei Q forme un treillis pour la divisibilité, pouvant être mis en bijection avec $\mathbb{B}^n$ si n est le nombre de facteurs premiers de Q. Application à Q = 30 030.

4.9.6

On considère l'ensemble des fonctions f de X à Y. On pose

$$f \propto g \leftrightarrow (\forall x \in \mathrm{dom}(f)\ g(x) = f(x))$$

Montrer que la relation $\propto$ est une relation d'ordre sur l'ensemble des fonctions de X à Y.

Montrer que cet ensemble a pour éléments maximaux les applications de X à Y, et pour élément minimum, la fonction définie en 0 point de X.

On dit f et g h-compatibles si $h \propto f$ et $h \propto g$. Montrer qu'il existe pour tout couple (f, g) une telle fonction. Définir pour chaque couple (f, g) une fonction h maximale pour $\propto$, et en déduire que $\propto$ dote l'ensemble des fonctions de X à Y d'une structure d'inf-demi-treillis.

On dira f et g h-miscibles si $f \propto h$ et $g \propto h$. Montrer qu'une telle fonction h n'existe pas toujours, et que $\propto$ ne dote pas l'ensemble des fonctions de X à Y d'une structure de treillis.

Interprétation technologique

Reprendre l'énoncé en considérant maintenant que les fonctions considérées désignent soit des spécifications, soit des réalisations, et que $f \propto g$ se lit « f spécifie g » ou « g réalise f ».

En déduire une notion de spécification progressive.

Discuter la notion de composants
- conformes à une spécification,
- compatibles entre eux.

Peut-on en général définir une réalisation r pouvant remplacer indifféremment 2 réalisations antérieures f et g, elles-mêmes compatibles avec une même spécification h ?

Application
- à l'univers des compatibles PC ,
- à l'univers des bibliothèques numériques.

4.9.7

On définit sur $\mathbb{R}$ les relations :

$x << y$	$\equiv$	$100\,\lvert x\rvert < \lvert y\rvert$	négligeabilité
$x \approx y$	$\equiv$	$\lvert x-y\rvert << \lvert y\rvert$	quasi-égalité
$x \sim y$	$\equiv$	$\neg(x << y) \wedge \neg(y << x)$	comparabilité

Montrer que la négligeabilité est un ordre strict.
Montrer que la quasi-égalité est une analogie plus fine que la comparabilité.
Montrer que ces analogies ne sont pas des équivalences.

4.9.8

Le jury de 9e année de l'Institut des Etudes Informatiques Internationales Interminables est agité par de graves débats : « il ne faut diplômer que les étudiants qui ont la moyenne (arithmétique) » disent les uns, « il faut favoriser les meilleurs », disent les directeurs de laboratoires ; le responsable du placement indique que tous peuvent être diplômés s'ils n'ont pas de faiblesse précise.

Pendant que les débats s'éternisent, le directeur des études s'inquiète. Le professeur Stanislas Fumet, parti à l'étranger, a laissé un logiciel de classement pour traiter les notes saisies, logiciel sensé pouvoir s'aligner sur toute politique du jury.

Au lancement, un écran rappelle que les étudiants seront classés à partir de leur 3 notes principales x, y, z dans l'ordre des $(x^n + y^n + z^n)^{1/n}$. Le logiciel demande la valeur du *paramètre* n.

Expliquer l'effet des différents n sur l'échantillon ci-après (tab. 4.12).

En déduire un étiquetage « politique » des différentes valeurs de n, pouvant servir d'aide-mémoire au Directeur des Etudes, ou de menu pour le logiciel.

4.9.9

On appelle *intervalles* des couples [a b] avec a,b $\in \mathbb{R}$ et $a \leq b$.

Définir et examiner une extension de $\leq$ et de = aux intervalles.

Introduire les comparateurs *coupe* et *dans*($\subseteq$), et examiner leurs propriétés.

Etudier ces divers comparateurs dans les 6 cas principaux de confrontation de [a b] à [c d].

Tableau 4.12 Classements.

nom	x	y	z	rang pour n =							
				−1	½	1	2	3	5	10	30
arthur	17	17	17								
ingmar	10	15	20								
gertrude	5	5	5								
pierrot	5	10	15								
roland	10	10	10								
zoé	15	15	15								

4.9.10 Choix d'une voiture

Pour préparer un logiciel sur ce thème, vous considérez :

- qu'un fichier vous définit des descriptifs standardisés des modèles connus,
- qu'il est basé sur divers critères, chacun ayant finalement ses valeurs ramenées à *une échelle standard d'intérêt* { - - -, - -, -, 0,+,++,+++}.

On se trouve par exemple ramené au tableau comparatif suivant.

Tableau 4.13 Comparatif de voitures.

Marque / modèle	prix d'achat	consom-mation	vitesse max	nb de places
Automobil Skorpio	++	– –	++	+
Behindra Victoria	+++	– – –	0	++
Cromwell Straight	+	–	+	+
Deleatur	0	0	+	–
Excalibur	– – –	– –	+++	++
Facel Vega	– –	– – –	++	+

1. Quelles sont les solutions dominantes ? dominées ?

On suppose qu'en fait tout utilisateur a ses priorités, et qu'il les ordonne avant de procéder à un choix[3].

2. Classer les solutions pour un *père de famille nombreuse* dont les priorités sont : nombre de places / prix d'achat / consommation / vitesse max, exploitées dans un ordre lexicographique sur les profils.

3. Même question pour un *jeune cadre* dont les priorités sont : vitesse max / prix à l'achat / consommation / nb de places.

4. Expliquer les différences avec les sélections faites sur la base :

(a) d'une note moyenne, tous les critères étant de même poids,

(b) d'une moyenne pondérée 8/4/2/1 ? 512/64/8/1 ? 4/3/2/1 ?

[3] Phase de méta-décision (J. Viallaneix).

4.9.11

Deux groupes représentatifs de deux populations ont donné leur ordre de préférence pour des émissions de télévision. Le groupe A préfère (par ordre décroissant) a, b, c, d, e, f, g, h ; le groupe B préfère, par ordre décroissant, a, d, e, f, b, c, h, g.

En déduire l'ordre partiel compatible avec ces 2 ordres totaux.

Peut-on ou non le considérer comme *plus objectif* que les ordres de préférence des groupes A et B ?

4.9.12

On souhaite classer les pays ayant participé aux Jeux Olympiques, sachant que chacun d'eux a remporté

- k médailles d'or,
- l médailles d'argent
- m médailles de bronze.

Peut-on bâtir un classement à partir d'un ordre unanime sur les triplets (k, l, m) ? à partir d'un ordre lexicographique (si oui, lequel ?).

On affecte maintenant 3 poids K, L, M aux trois nombres de médailles, et on classe les pays d'après leur score $K \cdot k + L \cdot l + M \cdot m$.

Un classement raisonnable suppose-t-il $K \geq L \geq M \geq 0$? $K > L > M > 0$?

Montrer que, dès que $M>0$, on obtient le même résultat avec (K, L, M) et (K/M, L/M, 1).

A quelle(s) condition(s) sur K, L, M retrouve-t-on l'ordre lexicographique ?

4.10 THÈMES DE RÉFLEXION

4.10.0 Sur le temps

Afin de pouvoir raisonner dans le temps, on utilise le modèle suivant. On considère un ensemble T de points représentant les temps possibles.

On définit sur T une relation R avec la signification

$$t1 \ R \ t2 \equiv t2 \text{ appartient au futur de } t1.$$

Commenter les propositions suivantes :

1. le temps du prévisionniste ou du futurologue est linéaire à gauche :

$$\forall t,u,s \ (tRs \wedge uRs) \rightarrow (tRu \vee t = u \vee uRt)$$

(i.e.: le passé est linéaire, mais le futur peut être arborescent)

2. le temps de l'historien ou de l'archéologue est linéaire à droite :

$$\forall t,u,s \ (sRt \wedge sRu) \rightarrow (tRu \vee t = u \vee uRt)$$

(le futur supporte l'arbre des passés possibles)

3. l'escatologie connaît un temps ultime ω :

$$\exists! \ t(\neg(\exists s \ tRs))$$

4. il y eut un premier temps α

$$\exists! \ t \ (\neg(\exists s \ sRt))$$

5. le temps des physiciens est totalement linéaire :

$$\forall t,u \ (tRu \vee t = u \vee uRt)$$

6. le temps des physiciens est dense / continu :

$$\forall t,s \ (tRs) \rightarrow \exists u \ (tRu \wedge uRs)$$

7. le temps des systèmes synchrones est discret :

$$\forall t,u,s \ (tRs) \leftrightarrow (t \ \rho \ s) \vee \exists u \ (tRu \wedge uRs)$$

$$\forall t,s \ (t\rho s) \rightarrow \neg\exists u \ (t\rho u \wedge u\rho s)$$

Relever les propriétés qui s'excluent.

Donner les principes du temps tel que vous le voyez, et les propriétés de la relation R associée.

4.10.1 Extension de la récurrence

Si une propriété P est vraie pour tous les nombres premiers, montrer que P est vrai dans tout $\mathbb{N}$ si

$$\forall x, y \in \mathbb{N}\ (P(x) \wedge P(y)) \rightarrow P(x \cdot y)$$

A-t-on aussi le droit de dire que P est vrai dans tout $\mathbb{N}$ à partir de

$$\forall x, y \in \mathbb{N}\ (P(x) \wedge P(y)) \rightarrow P(\mathrm{ppcm}(x, y))\ ?$$

4.10.2 Théorie(s) des passoires

Evaluer et comparer les théories suivantes.

THÉORIE 1. Les passoires sont des ustensiles ayant en général un grand nombre de trous ; de fait, si je considère une passoire, et que je double son nombre de trous, j'ai encore une passoire ; et si je divise son nombre de trous par deux, j'ai aussi une passoire ; d'où :

PROPOSITION. Une passoire est un ustensile ayant en général un grand nombre de trous, ce nombre de trous n'ayant pas d'importance particulière (Pr Shadoko, in Jacques Rouxel, *Les Shadoks*, Edition Circonflexe, 1994).

THÉORIE 2. Les passoires sont des ustensiles ayant un certain nombre de trous ; de fait, ce nombre de trous sert à les ordonner, des moins trouées aux plus trouées. Récemment, quelques auteurs ont posé la question d'assimiler les casseroles à des passoires à 0 trou, mais on peut légitimement se demander si ce genre de contorsions est utile au progrès des sciences.

THÉORIE 3. Les passoires sont des ustensiles ayant un certain nombre de trous ; de fait, on pose p –: q (lire p « pas plus troué que » q) si le nombre de trous de la passoire p n'excède pas celui de q.

La relation –: est ainsi un *préordre large et total* sur l'ensemble P des passoires. On notera P_k la classe d'équivalence regroupant les passoires à k trous.

Les passoires au sens strict supposent $k > 0$: dans ce cadre, la classe minimale ou classe des passoires les moins trouées est alors la classe P_1 dite classe des *entonnoirs*.

Dans le cadre plus général de la théorie des ustensiles, on peut admettre que P est minoré par une classe P_0 de passoires au sens large, ou passoires à 0 trou, usuellement appelées *casseroles*.

Envisager un prolongement de la théorie des passoires aux *tamis* et aux *filtres*.

4.10.3 Théorie de la modélisation

On considère un système réel S, et la famille $M = \{M_i\}$ des modèles bâtis par le ou les observateur(s) à partir des observations & connaissances à son sujet.

Soit une situation définie par le couple E = (situation interne, état de l'environnement). On appelle R(E) l'ensemble des conséquences observées si S est mis dans l'état E, et $P_i(E)$ les conséquences prévues selon le modèle M_i avec la même situation initiale. On définit le *domaine de validité* de M_i par :

$$\mathrm{dom}(M_i) = \{E \mid R(E) \approx_i P_i(E)\}$$

où $\approx_i$ désigne une relation d'équivalence attachée au modèle.

Montrer que $M_i \angle M_j \leftrightarrow (\mathrm{dom}(M_i) \subseteq \mathrm{dom}(M_j) \wedge (\approx_i \supseteq \approx_j))$ définit un préordre (large) de validité entre modèles à partir

- de l'inclusion entre leurs domaines,
- de la finesse entre leurs équivalences.

Soit k(m) un indice de complexité du modèle m.

Montrer que $M_i << M_j \leftrightarrow M_i \angle M_j \wedge k(M_i) \geq k(M_j)$ constitue un préordre(large) de pertinence.

Application. Expliquer, par exemple, en matière d'astronomie, le passage du modèle de Ptolémée au modèle de Newton.

Peut-on avoir pour un système un modèle parfait ?

4.11 LECTURES

AHO A., ULLMAN J., *Concepts fondamentaux de l'informatique*, chap. VIII et IX, Dunod, 1993.

ARROW K.J., *Social Choice and Individual Values,* Wiley, New York, 1951, 1963... trad : *Choix collectif et Préférences individuelles*, Calmann-Lévy, 1974.

FAURE R., HEURGON E., *Structures ordonnées et algèbres de Boole,* Coll. Programmation, Ed. Gauthier-Villars, 1971.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien,* tome I, chap. II, Coll. Programmation, Ed. Gauthier-Villars, 1973.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science,* vol. I, chap. II & III, Coll. Texts and Monographs on Computer Science, Springer Verlag, 1991.

KAUFMANN A., BOULAYE G., *Théorie des treillis en vue des applications*, chap. I, Ed. Masson, 1978.

REINHARDT R. & SOEDER H., *Atlas des mathématiques,* chap. 2 et 3, La Pochothèque, Librairie Générale Française, 1997.

SCHÄRLIG A., *Décider sur plusieurs critères*, Coll. Diriger l'Entreprise, Presses polytechniques et universitaires romandes, 1985/1990.

Collectif, *Petite encyclopédie des mathématiques,* chap. XIV, Edition Pagoulatos, Didier distr., 1980.

CHAPITRE 5

FONCTIONS, CALCULABILITÉ, RÉCURRENCE

[The Analytical Engine is limited to]
whatever we know about how to order it to perform.
Ada Lovelace, d'après Charles Babbage (~1835)

Les années 1940 ont introduit des machines programmables – les futurs ordinateurs – semblant permettre d'effectuer tous calculs sur des entiers, et notamment des calculs paramétrables établissant la valeur de fonctions. De là, « à un codage près », on pouvait mener des calculs arbitraires, menant à des fonctions de toutes sortes.

Cependant, dès les années 1930, quelques chercheurs s'étaient occupés d'identifier scientifiquement ce qui serait calculable, effectivement ou potentiellement, et ce qui ne le serait pas.

Ce chapitre donne des indications sur une tripartition des fonctions au sens le plus général en

- fonctions arithmétiques directement calculables,
- autres fonctions calculables « à un codage près »,
- autres fonctions, conceptuellement définies sans être calculables.

Informatique et mathématiques partagent les deux premières classes, tandis que la dernière montre les limites mathématiques de l'informatique.

5.1 CALCULABILITÉ, DÉCIDABILITÉ

DÉFINITIONS

5.1 – Une ***fonction*** sera dite ***calculable*** s'il existe un moyen ou procédé déterministe permettant d'établir sa valeur pour n'importe quel argument en une durée finie – ou, plus généralement, à l'aide de ressources finies[1]. De même, une fonction pourra être ***partiellement calculable*** voire ***non calculable.***

5.2 – On appellera ***prédicat*** une fonction à valeur dans {0, 1} servant à trancher une question.

5.3 – Une ***question*** sera dite ***décidable*** si le prédicat associé est calculable : alors, il existe un moyen ou procédé déterministe permettant d'établir la réponse par oui ou non, pour n'importe quel argument en une durée finie – ou, plus généralement, à l'aide de ressources finies.

---

[1] C'est une version forte ; une version faible se contenterait d'un procédé ayant un nombre fini d'étapes, ce qui revient au même si chacune de ses étapes est elle-même finie.

5.4 – Une ***question*** sera dite ***semi-décidable*** s'il existe un moyen ou procédé déterministe permettant d'établir une réponse oui (resp. une réponse non) pour n'importe quel argument en une durée finie – ou, plus généralement, à l'aide de ressources finies ; autrement dit si le prédicat associé est partiellement calculable.

REMARQUE. Typiquement, la séquence 327 appartient-elle à l'écriture décimale d'un nombre transcendant tel que e ? On peut engendrer les n premières décimales de e, et voir facilement, dans un temps fonction de n, si 327 appartient à cette séquence. Une réponse affirmative à cette question est une réponse affirmative à la première. Par contre, une réponse négative sur les n premiers chiffres suggère simplement de pousser les tests au-delà du $n^{ième}$ chiffre. L'appartenance de 327 à l'écriture décimale d'un nombre transcendant tel que e peut être considérée comme typiquement semi-décidable : si, à un moment donné, « oui » n'est pas atteint, il faut poursuivre pour savoir si oui ou non. Si bien que « non » ne peut être atteint en une durée finie.

5.2 CALCULABILITÉ ET RÉCURSIVITÉ

5.2.1 Les thèses fondamentales

Alonzo Church a postulé dès 1936 que :

THÈSE 5.1 – Toute fonction calculable (au sens large ci-dessus) est récursivement calculable (au sens étroit ci-après).

THÈSE 5.2 – Toute fonction récursivement calculable est calculable sur une machine de Turing.

Ce qu'on complète généralement par le postulat 5.3.

POSTULAT 5.3 – Tout « calcul » sur une machine de Turing peut être reproduit par un « calcul » sur ordinateur, et réciproquement.

Ce postulat, jamais battu en brèche, est d'autant plus fort que la machine de Turing est rudimentaire.

5.2.2 Machine de Turing

En 1936, Alan Turing définit une machine conceptuelle simple, surtout efficace par ses preuves négatives : si on montre sur le papier qu'il n'existe pas de machine de Turing pertinente pour un problème, alors il n'existe pas d'ordinateur constructible qui puisse l'être.

Une machine de Turing comprend un ruban et un automate. Le ruban est divisé en cases contenant un symbole (ou un blanc). L'automate, caractérisé à tout moment par son état, travaille pas à pas, et possède une tête de lecture/écriture sur le ruban. Au départ, puis à chaque pas jusqu'à l'arrêt :

- l'automate lit le symbole s du ruban situé sous sa tête de lecture,
- en fonction de son état interne i et de ce symbole s
  - éventuellement, il écrit un symbole s′ sur le ruban,
  - éventuellement, il avance ou recule le ruban d'une case,
  - éventuellement, il passe dans un état i′,

ce qu'on résume par un jeu de règles du genre

$$(s, i) \rightarrow (s', i', m), \text{ avec } m \in \{\text{gauche}, _, \text{droite}\}$$

Ces règles peuvent être rassemblées dans un tableau R, à raison d'une ligne par état i, une colonne par symbole s ; la règle ci-dessus se code par

$$R[i, s] = (s', i', m)$$

Similaire aux automates à états finis du chapitre 16, la possibilité pour la machine de Turing d'utiliser le ruban comme mémoire auxiliaire la rend incomparablement plus puissante.

EXEMPLE. On considère une machine de Turing M, dont le ruban comporte deux suites de lettres terminées par un #.

On suppose la tête de lecture en T, et la machine dans l'état 0. Avec quels états et quelles règles *la machine M peut-elle décider si les deux suites sont identiques* ?

On emploie une méthode destructrice : la comparaison de *baobab#* à *baba#* est ramenée à celle de aobab# à aba#, puis à celle de obab# à ba#, différentes puisque o≠b ; la comparaison de boa# à boabdil# est de même ramenée à celle de # à bdil#, différentes puisqu'une suite est vide avant l'autre. L'identité est obtenue s'il reste à comparer # à #.

Tableau 5.1 Suite d'états du ruban pour M.

T				suite1							suite2					début
			b	a	o	b	a	b	#		b	a	b	a	#	phase 0
		b	a	o	b	a	b	#		b	a	b	a	#		suite1 ?
	b	a	o	b	a	b	#		b	a	b	a	#			
b	a	o	b	a	b	#		b	a	b	a	#				phase 1
	a	o	b	a	b	#		b	a	b	a	#				b 1?
a	o	b	a	b	#		b	a	b	a	#					
o	b	a	b	#		b	a	b	a	#						
b	a	b	#		b	a	b	a	#							
a	b	#		b	a	b	a	#								
b	#		b	a	b	a	#									
#		b	a	b	a	#										b2 ?
	b	a	b	a	#											
b	a	b	a	#												b !
	a	b	a	#												retour
		a	b	a	#											suite 1 ?
#			a	b	a	#										retour
b	#			a	b	a	#									tête
a	b	#			a	b	a	#								
b	a	b	#			a	b	a	#							
o	b	a	b	#			a	b	a	#						
a	o	b	a	b	#			a	b	a	#					
	a	o	b	a	b	#			a	b	a	#				phase 0
a	o	b	a	b	#			a	b	a	#					suite1
	o	b	a	b	#			a	b	a	#					a ?
o	b	a	b	#			a	b	a	#						
b	a	b	#			a	b	a	#							

Affectons l'état 0 au défilement du ruban vers la gauche, à la recherche du premier symbole de suite 1 ; ce défilement se poursuit tant que la tête T voit un blanc (noté Δ par la suite).

Au premier symbole (b) de la suite 1, M passe dans un état (b) mémorisant la lettre, et l'efface (baobab# devient aobab#). M fait défiler le ruban en restant dans cet état

jusqu'au #, puis passe alors dans l'état destructeur associé (b → B), qui reste stable tant qu'il voit un caractère blanc. M aborde ensuite la deuxième suite.

- Si M est dans l'état # et trouve #, les deux suites étaient identiques (état 98).
- Si M est dans un autre état X (resp. B) et trouve une lettre x compatible (resp. b), il l'efface, (baba# devient aba#) passe dans un état 1 de recherche de la première suite, puis au # intercalaire, passe dans un état 2 de recherche de la tête : la suite 1 dépassée (Δ) le remet dans l'état 0. On compare maintenant aobab# à aba#...
- Si M est dans un état X et trouve une lettre ou un # non compatible, les deux suites n'étaient pas identiques (état 99).

Tableau 5.2 Table de transition de M ; cas stables (*tant que*) en gras.

	états	Δ	a	b	..	o	...	#
recherche début suite 1 à droite	0	**0,,g**	a, Δ, g	b, Δ ,g		o, Δ ,g		#
retour vers suite 1	1	**1,,d**	ω	ω		ω		2,,d
recherche tête suite 1 à gauche	2	0,,g	**2,,d**	**2,,d**		**2,,d**		ω
a trouvé en tête de suite 1	a	ω	**a,,g**	**a,,g**		**a,,g**		A,,g
b trouvé en tête de suite 1	b	ω	**b,,g**	**b,,g**		**b,,g**		B,,g
a en tête de suite 2 ?	A	**A,,g**	1,Δ,d	99,,g		99,,g		99,,
b en tête de suite 2 ?	B	**B,,g**	99,,	1,Δ,d		99,,		99,,
# en tête de suite 2 ?	#	**#,,g**	ω	ω		ω		98,,
halte /oui	98							**98,,**
halte /non	99	**99,,g**	**99,,g**	**99,,g**		**99,,g**		**99,,**
erreur	ω	**ω,,g**	**ω,,g**	**ω,,g**		**ω,,g**		**ω,,g**

5.2.3 Fonctions élémentaires

DÉFINITION 5.5 – Une ***fonction élémentaire*** a pour définition :

- soit une constante, comme pour les traditionnelles fonctions zéro, de définition : $\text{zéro}_D : D \rightarrow 0$, où D est un domaine quelconque ;
- soit un des arguments de la fonction : on appelle parfois *projection* les fonctions $I^k_n(x_1, \ldots x_n) \equiv x_k$, (où $k \leq n$), telles que *tête:* $(x_1, x_2, x_3) \rightarrow x_1$;
- soit la fonction successeur, définie par $\text{succ}(n) \equiv n + 1$.

5.2.4 Fonctions composées

On étend la notion de composition de fonction : f o g : x → f(g(x)) aux cas tels que : k(x1,x2,x3) ≡ r(f(x1, x2, x3), g(x1, x2, x3), h(x1, x2, x3)).

C'est le cas des fonctions définies par une *expression ou formule* :

$$h(x1, x2, x3) \equiv (x1 \cdot x2 + x1 \cdot x3),$$

qu'on peut assimiler à :

$$h(x1, x2, x3) \equiv add(mult(x1, x2), mult(x2, x3))$$

(chap. 12, partie Algèbre), ou

$$h(x1, x2, x3) \equiv add(mult12(x1, x2, x3), mult23(x1, x2, x3)).$$

C'est aussi le cas de fonctions arbitraires, mises en correspondance avec des fonctions de $\mathbb{N}^{\mathbb{N}}$ par des fonctions de codage/ représentation $r_X : X \to \mathbb{N}$ et des fonctions de décodage/ interprétation $i_Y : \mathbb{N} \to Y$:

$$\begin{array}{ccc} & r_X & \\ & X \to \mathbb{N} & \\ g \downarrow & & \downarrow f \\ & Y \leftarrow \mathbb{N} & \\ & i_Y & \end{array}$$

Fig. 5.1 Calcul « à un codage près ».

$$g : X \to Y \text{ est } \textit{réalisée par } f \text{ si } g \equiv i_Y \text{ o } f \text{ o } r_X \text{ (fig. 5.1)}$$

5.2.5 Conditionnelle de McCarthy

DÉFINITION 5.6 – On appelle ***expression conditionnelle***[2] une expression formellement notée :

$$\{ c1 \to e1 \; ; \; c2 \to e2 \; ; \; \ldots \; ; \; cn \to en \}$$

qui se lit

si c1 alors e1 sinon si c2 alors e2 sinon...sinon si cn alors en

où les c_i sont des conditions, les e_i des expressions.

NOTES

- On supposera en général {c1∨ c2 ∨...∨ cn} vraie pour que l'expression soit totalement définie.
- On supposera les conditions disjointes 2 à 2 si on veut garantir l'unicité de définition, et aussi que les branches *ck → ek*, supposées en nombre fini, sont permutables.

5.2.6 Fonctions à domaine fini

PROPOSITION 5.1 –Toute fonction définissable totalement par une table (tab. 5.1) peut être définie par une conditionnelle de MacCarthy à un seul niveau :

$$f(x) \equiv \{x = A \to P \; ; \; x = B \to Q \; ; \; x = C \to R \; ; \; x = D \to S\}$$

n'utilisant que des tests d'identité comme préconditions, et des constantes comme valeurs.

Le dernier cas étant résiduel, il peut être indifférent d'écrire :

$$f(x) \equiv \{x = A \to P \; ; \; x = B \to Q \; ; \; x = C \to R \; ; \; \to S\}$$

voire $f(x) \equiv \{x = A \to P \; ; \; x = B \to Q \; ; \; x = C \to R \; ; \; S\}$

si f est totalement définie ($\neg \exists x \notin \{A,B,C,D\}$) : tableau 5.3.

Tableau 5.3

Argument	Valeur
A	P
B	Q
C	R
D	S

[2] Concept introduit en 1959 par John McCarthy dans son langage LISP, et survivant en C, en Scheme...

5.2.7 Evaluation partielle

Au-delà d'un argument, on peut définir de même une fonction en partant de ses évaluations partielles relatives à un argument à domaine fini, en posant par exemple :

$$f(x,y) \equiv \{x = a \rightarrow f_a(x) \; ; \; x = b \rightarrow f_b(y); \; x = c \rightarrow f_c(y) \; ; \; x = d \rightarrow f_d(y) \}$$

5.2.8 Fonctions à conditionnelle générale

Considérons des définitions conditionnelles générales telles que :

$$\text{Fenêtre}(x, y, z) \equiv \{ x<y \rightarrow y \; ; \; x>z \rightarrow z \; ; \; x)$$
$$\text{Sel}(n,f,g,h,x) \equiv \{n = 0 \rightarrow f(x), \; n = 1 \rightarrow g(x), \; n = 2 \rightarrow h(x)\}$$

PROPOSITION 5.2 – On ne sort pas du domaine des fonctions calculables si une fonction est définie par une conditionnelle renvoyant à des conditions décidables et des expressions calculables.

En particulier, on peut vérifier par récurrence que ce sera le cas si, partant d'une famille initiale f_0, ... f_i de fonctions à domaine fini, définies comme ci-dessus, on l'étend à une suite $\{f_j\}$ de fonctions qui restent toutes calculables *du seul fait que leurs conditionnelles n'utilisent que les fonctions qui les précèdent.*

Par contre, considérons la fonction « non traçable » définie par Dirichlet en 1828 :

$$f(x : \text{réel}) \equiv \{\text{estRationnel}(x) \rightarrow 1 \; ; \; 0\}$$

Cette fonction n'est pas calculable car le prédicat estRationnel(x) qu'elle reproduit n'est pas décidable.

Considérons le cas d'un réel représenté en numération de position en base B. S'il est entier, à une puissance de la base près, sa représentation est finie. Si c'est un autre rationnel, un nombre algébrique, un nombre transcendant, sa représentation est infinie. Ce qui distingue les rationnels est le fait que leur représentation est alors formée d'une suite de chiffres quelconque, suivie d'une suite de chiffres infinie périodique. Mais s'il faut tester cette représentation :

- en commençant par le début, on ne peut déterminer où se trouve l'éventuelle transition entre les deux suites, l'apparition d'un motif périodique pouvant être un leurre local ;
- la représentation étant infinie, il est dur de commencer par la fin !

De fait, l'informatique représente (approximativement) les réels par des rationnels, rendant estRationnel(x) indécidable à partir des représentations.

5.2.9 Définition récursive

DÉFINITION 5.7 – On parlera de ***composition récursive*** si une fonction peut s'utiliser elle-même dans sa définition, directement ou par l'intermédiaire d'autres fonctions.

Ce qui revient à dire que toute fonction f_i est habilitée à utiliser dans sa définition les fonctions f_0 à, disons, f_{i+20}.

Cette ***circularité de définition*** est-elle maléfique ou bénéfique ?

5.3 RÉCURSIVITÉS CONVERGENTE, DIVERGENTE, STATIONNAIRE

5.3.1 Convergence

DÉFINITION 5.8 – Une ***définition récursive*** sera dite ***convergente***, quand, malgré sa(ses) circularité(s), elle est évaluable en un nombre fini d'étapes.

Pour cela, on distingue dans les conditionnelles

- les cas échappant à la circularité, et qui forment la ***plage d'arrêt*** ;
- d'autres cas utilisant cette circularité apparente, et dont on montrera par récurrence[3] qu'ils se ramènent tôt ou tard – toujours en un nombre fini d'étapes – aux cas précédents.

On pourra utiliser les permutations de branches licites pour régler cette question, des cas les plus simples aux plus complexes. La réduction d'un cas général à un cas strictement plus simple sera toujours un bon indice, si ce n'est une preuve définitive.

EXEMPLES

- Soit la fonction définie par :

$$\text{fact}(n) \equiv \{\, n = 0 \rightarrow 1 \;;\; \text{fact}(n-1) \cdot n \,\}$$

  Si cette fonction ne reçoit que des arguments positifs :
  - ou c'est 0, et le calcul est fini en 1 étape,
  - ou n est quelconque, et fact(n) appelle fact(n – 1), qui est fact(0) ou appelle fact(n – 2) ... processus convergent vers l'appel de fact(0).

  Plus précisément, si app(n) est le nombre effectif d'appels de fact() par fact(n), alors app(n) = {n = 0 → 1 ; 1 + app(n – 1)} ; par récurrence, app(n) = n + 1 ; le calcul de fact(n) comporte finalement n appels successifs à fact(), l'évaluation de fact(0) puis les n multiplications, soit 2n + 1 étapes élémentaires.

- Soit la fonction définie par

$$\text{fib}(n) \equiv \{n = 0 \vee n = 1 \rightarrow n \;;\; \text{fib}(n-1) + \text{fib}(n-2)\}$$

  - fib(0) et fib(1) s'évaluent en une étape ;
  - si, pour $k \geq 1$, les fib(0) à fib(k) s'évaluent en un nombre fini d'étapes, il en est de même pour fib(k) + fib(k – 1), c'est à dire pour fib(k + 1) ;

  la récursivité est donc convergente (par récurrence).

- Recettes Logo : en considérant la hiérarchie phrase/ mot/ lettre, il vient :
  - pour écrire une phrase, si elle n'est pas vide : écrire le premier mot, un blanc, puis le reste de la phrase ;
  - pour écrire un mot, s'il n'est pas vide : écrire la première lettre puis le reste du mot .

  A quelles conditions ces 2 définitions sont-elles convergentes ?

5.3.2 Divergence

DÉFINITION 5.9 – Une ***définition récursive*** sera dite ***divergente*** s'il existe des situations où l'on s'écarte irrésistiblement de la plage d'arrêt, ce qui se manifestera en programmation par un comportement « explosif ».

---

[3] Le terme de *récurrence* sera gardé pour les raisonnements, et celui de *récursivité* pour les définitions de fonction : la récurrence – usant d'une ou deux règles d'inférence – servira à juger des propriétés des récursivités.

EXEMPLES

- Soit la fonction : $\text{fact}(n) \equiv \{n = 0 \rightarrow 1 \; ; \; n \cdot \text{fact}(n-1)\}$.

  Si cette fonction reçoit – 1, cela provoque des appels pour – 2, – 3... sans aucune limite (divergence abyssale).

- Soit la fonction $\text{fib}(n) \equiv \{n = 0 \vee n = 1 \rightarrow n \; ; \; \text{fib}(n-1) + \text{fib}(n-2)\}$

  Essayons de l'accélérer : $\text{fib}(n-1) = \text{fib}(n-2) + \text{fib}(n-3)$; posons :

  $$\text{fib}(n) \equiv \{n = 0 \vee n = 1 \rightarrow n \; ; \; 2 \cdot \text{fib}(n-2) + \text{fib}(n-3)\}$$

  Dans cette nouvelle formulation, fib(2) engendrera des appels successifs à fib(– 1), fib(– 4)... ne pouvant mener à la plage d'arrêt, d'où divergence ; fib(3) converge, mais au-dessus la divergence réapparaît : fib(4) et fib(5) emploient fib(2), fib(6) et fib(7) emploient fib(4)...

  La plage d'arrêt devient passoire. Pour éviter ces évasions, il faut « boucher le trou » et écrire :

  $$\text{fib}(n) \equiv \{n = 0 \rightarrow 0, n = 1 \vee n = 2 \rightarrow 1 \; ; \; 2 \cdot \text{fib}(n-2) + \text{fib}(n-3)\}$$

- Soit la fonction $f(n) \equiv \{n \leq 0 \rightarrow 0 \; ; \; 1 + f(\lceil n/2 \rceil)\}$.

  Pour n>0, on atteint f(1) dont l'évaluation $f(1) = 1 + f(1) = 1 + (1 + f(1)) = 1 + (1 + (1 + f(1)))$... diverge.

  Par contre, la fonction $f(n) = \{n \leq 0 \rightarrow 0 \; ; \; 1 + f(\lfloor n/2 \rfloor)\}$ converge sans problème : $f(1) = 1 + f(0) = 1$.

5.3.3 Stationnarité

DÉFINITION 5.10 – Une ***définition récursive*** sera dite ***stationnaire*** (resp. circulaire) si elle engendre une situation (resp. un cycle) se reproduisant indéfiniment.

Il est clair :

- qu'il n'y a pas divergence ;
- qu'il n'y a pas arrêt, en ce sens que la situation atteinte (resp. le cycle atteint) échappe(nt) à la plage d'arrêt ;
- plus qu'un comportement « explosif », il y a donc lieu de craindre un interminable silence !

EXEMPLES

- $\text{silence}(n) = \{n = 0 \rightarrow 0 \; ; \; \text{silence}(n)\}$
- $\text{bascule}(n) = \{n = 0 \rightarrow 0 \; ; \; \text{bascule}(-n)\}$

En termes de programmation, le seul espoir est qu'il n'y ait pas d'optimisation, et que la pile d'évaluation consente à déborder (en pratique, la divergence est parfois perçue comme un moindre mal, car *révélatrice d'un problème*).

5.3.4 Calculabilité, semi-calculabilité

DÉFINITION 5.11 – Sera dite ***calculable*** toute fonction dont l'évaluation réussit en un temps fini pour chaque jeu d'arguments.

PROPOSITION 5.3 – Seront calculables :

- les fonctions élémentaires,

- les expressions conditionnelles dont les conditions c sont décidables, et les expressions e attachées calculables,
- les fonctions non récursives n'utilisant que des fonctions calculables,
- les fonctions récursives convergentes sur la totalité de leur domaine.

Une fonction *semi-calculable* pourra être stationnaire ou divergente pour certaines valeurs pour lesquelles elle n'est pas définie.

En pratique, on préférera une définition calculable, émettant un symbole défini (par exemple ω) en cas d'erreur[4] (fonctions « blindées »), ce qui signifie que les situations erronées sont maîtrisées.

Comparons :

- factN(n) : $\mathbb{N} \to \mathbb{N} \equiv \{ n = 0 \to 1 ; \text{factN}(n-1)\cdot n \}$
  calculable, récursive convergente sur tout $\mathbb{N}$;
- factZ0(n) : $\mathbb{Z} \to \mathbb{N} \equiv \{ n = 0 \to 1 ; \text{factZ0}(n-1)\cdot n\}$
  divergente pour $n<0$;
- factZ1(n) : $\mathbb{Z} \to \mathbb{N}\cup\{\omega\} \equiv \{ n<0 \to \omega ; n = 0 \to 1 ; \text{factZ1}(n-1)\cdot n\}$
  blindée.

Pour factZ1 le diagnostic de non-définition est immédiat.

Dans factZ0, à déconseiller, ce qui en tient lieu c'est une divergence en théorie illimitée, dont l'exécution serait en pratique bloquée par une erreur-système pas forcément récupérable.

5.4 ENSEMBLES RÉCURSIFS ET RÉCURSIVEMENT ENUMÉRABLES

5.4.1 Ensembles récursivement énumérables

DÉFINITION 5.12 – Les ***ensembles récursivement énumérables*** sont des ensembles dont un processus permet d'énumérer les éléments. Le processus est fini si l'ensemble est fini.

EXEMPLE. L'ensemble des décimales de π est récursivement énumérable : on sait engendrer ces décimales, mais elles constituent une suite infinie non périodique, π n'étant pas rationnel.

5.4.2 Ensembles récursifs

DÉFINITION 5.13 – Les ***ensembles récursifs*** sont des ensembles récursivement énumérables, et pour lesquels l'appartenance est décidable.

Il est crucial qu'un langage – considéré comme l'ensemble de ses productions – soit récursif, pour pouvoir décider si une phrase ou séquence donnée lui appartient ou non – dans le cas positif, on espèrera en sous-produit une analyse de la phrase testée.

---

[4] A rapprocher du symbole NaN (Not a Numeric value) pour les virgules flottantes (norme IEEE754).

5.5 AUX SOURCES DE L'ARITHMÉTIQUE ET DE L'ALGÈBRE

5.5.1 Ensemble $\mathbb{N}$ des entiers naturels

On considère un ensemble $\mathbb{N} = < 0 ; s>$, comportant un objet 0, et support d'une application $s : \mathbb{N} \rightarrow \mathbb{N}$, associant à un élément n son successeur s(n).

Peano a montré en 1891 que cette structure décrivait l'ensemble des entiers naturels aux conditions suivantes (*Petite encyclopédie des mathématiques*, p. 77)

- 0 est un nombre ;
- tout nombre n possède un et un seul successeur s(n) ;
- 0 n'est le successeur d'aucun nombre ;
- des nombres distincts ont des successeurs distincts ;
- $\mathbb{N}$ contient 0 et tout successeur d'un nombre lui appartenant.

On construit progressivement $\mathbb{N} = < 0, 1 ; s, p ; +, \cdot, \wedge >$ en introduisant d'abord le *prédécesseur* p(x) de x par la relation : p(s(y)) = y, et, si l'on veut, p(0) = ω ; puis :

$$x + y \equiv add(x, y) \text{ , avec } add(x,y) \equiv \{y = 0 \rightarrow x ; s(add(x, p(y))) ;$$

+ se révèlera associatif, commutatif, d'élément neutre 0…

$$x \cdot y \equiv mult(x, y), \text{ avec et } mult(x,y) \equiv \{ y = 0 \rightarrow 0 ; x + mult(x, p(y))\}$$

· se révèlera associatif, commutatif, d'élément absorbant 0, d'élément neutre s(0) = 1, distributif par rapport à + …

$$x \wedge y \equiv x^y \equiv e(x, y), \text{ avec } e(x, y) \equiv \{y = 0 \rightarrow 1 ; mult(x, e(x, p(y))\}$$

1 se révèlera neutre à droite, absorbant à gauche, donc idempotent…

5.5.2 Ensemble $\mathbb{Z}$ des entiers signés

Soient x et y, éléments de $\mathbb{N}$; x + y est toujours défini, mais existe-t-il toujours un z (noté x – y) tel que x = y + z ? Ce z n'appartient à $\mathbb{N}$ que si $x \geq y$. Sinon, « on sort des limites de l'épure ».

Les Grecs en avaient déduit que le résultat n'était pas défini… puisqu'ils n'en possédaient pas de définition.

Aux premiers siècles de notre ère, les Indiens convinrent que le problème pouvait se régler par analogie avec les avoirs et leur opposé les dettes :

grandAvoir – petitAvoir = avoir, petitAvoir – grandAvoir = dette

Au-delà, on pouvait s'intéresser aux « dettes – avoirs », aux « avoirs – dettes » et même aux « dettes – dettes », avec par exemple petiteDette – grandeDette = avoir.

De même, nous introduisons l'ensemble $\mathbb{Z}$ des entiers signés avec les règles :

- $x \in \mathbb{N} \rightarrow x \in \mathbb{Z}$
- $x \in \mathbb{Z} \rightarrow -x \in \mathbb{Z}, x + (-x) = 0$
- $x \in \mathbb{Z}, y \in \mathbb{Z} \rightarrow x + y \in \mathbb{Z}$
- $x \in \mathbb{Z}, y \in \mathbb{Z} \rightarrow x - y \equiv x + (-y)$

$\mathbb{Z}$ constitue un ***symétrisé*** de $\mathbb{N}$ pour l'addition.

C'est le plus petit ensemble permettant de toujours répondre positivement à la question « si x et y sont dans $\mathbb{N}$, existe-t-il toujours un z (noté x – y) tel que x = y + z ? ».

De plus, « si x et y sont dans $\mathbb{Z}$, il existe toujours dans $\mathbb{Z}$ un z noté x – y tel que x = y + z ».

Toute arithmétique sur $\mathbb{Z}$ peut se ramener à une arithmétique sur $\mathbb{N}$, à l'aide de la bijection définie par

$2n - 1 \in \mathbb{N} \rightarrow -n \in \mathbb{Z}$, $2n \in \mathbb{N} \rightarrow n \in \mathbb{Z}$ de $\mathbb{N}$ à $\mathbb{Z}$

et par

$n \in \mathbb{Z} \rightarrow (n \geq 0 \rightarrow 2n\ ;\ 2|n| - 1)$ de $\mathbb{Z}$ à $\mathbb{N}$

5.5.3 Ensembles $\mathbb{F}$ des fractions & $\mathbb{Q}$ des rationnels

Soient x et y de $\mathbb{N}$; $x \cdot y$ est toujours défini, mais existe-t-il toujours un z de $\mathbb{N}$ tel que $x = y \cdot z$? Rarement !

Introduisons de nouveaux objets dits ***fractions***, notés x / y ([5]), et tels que : $(x/y) \cdot y = x$ pour $y \neq 0$. Soit $\mathbb{F} = \{x/y \mid x \in \mathbb{N}, y \in \mathbb{N}\}$ leur ensemble. En posant x = x/1 et 1/(x /y) = y/x, le nouvel ensemble $\mathbb{F}$ est stable pour l'addition, la multiplication et la division. Il est également stable pour la soustraction si on l'étend aux fractions négatives, représentées par des couples de $\mathbb{Z} \otimes \mathbb{N}$.

Par construction, $\mathbb{F}$ *correspond bijectivement à* $\mathbb{N}^2$, et diverses techniques de numérotation permettent d'affirmer que $\mathbb{N}^2$ et $\mathbb{N}$ peuvent aussi être mis en bijection : donc, toute arithmétique sur $\mathbb{F}$ peut se ramener à une arithmétique sur $\mathbb{N}$.

Cependant, on sait depuis longtemps que ½ = 2/4 = 3/6 … et qu'ainsi un grand nombre de couples ou fractions sont équivalents. Soit $m \neq 0$, $p, q \neq 0$ 3 entiers : on a $p/q = (m \cdot p)/(m \cdot q)$. Posons $p/q = \mathrm{red}((m \cdot p)/(m \cdot q))$; nous dirons p/q ***réduite*** de $(m \cdot p)/(m \cdot q)$. Une réduite p/q est irréductible si p et q sont premiers entre eux : on ne peut plus alors faire surgir de facteur commun m permettant une nouvelle réduction, et l'on notera alors $(p/q) = \mathrm{red}^*(m \cdot p/(m \cdot q))$.

L'ensemble $\mathbb{Q}$ des rationnels se présente comme l'ensemble des classes d'équivalence sur $\mathbb{F}$. Concrètement, il regroupe les représentants nécessaires et suffisants des classes de $\mathbb{F}$, savoir les ***fractions irréductibles***. Ainsi, à tous les 0/x de $\mathbb{F}$ (avec $x \neq 0$) correspond le 0 de $\mathbb{Q}$. A tous les $x/(p \cdot x)$ ($x \neq 0$, $p \neq 0$) correspondent les quantièmes 1/p de $\mathbb{Q}$. En étendant $\mathbb{Q}$ aux fractions négatives, on peut finalement montrer que $\mathbb{Q}$ est stable pour l'addition, la multiplication, la soustraction et la division. $\mathbb{Q}$ supporte enfin un ordre total <, quotient du préordre < sur $\mathbb{F}$.

$\mathbb{Q}$ étant un sous-ensemble de $\mathbb{F}$, une arithmétique sur $\mathbb{Q}$ se présentera en général comme une arithmétique sur $\mathbb{F}$ gérant des données et des résultats « normalisés » au sens de la réduction ci-dessus. Ainsi,

$$\mathrm{add}_F\,(x/y, z/t) = (x \cdot t + y \cdot z)/(y \cdot t)$$

où x/y et z/t ne sont pas forcément réduites, tandis que

$$\mathrm{add}_Q\,(x/y, z/t) = \mathrm{red}^*((x \cdot t + y \cdot z)/(y \cdot t)) = \mathrm{red}^*(\mathrm{add}_F(x/y, z/t))$$

où x/y et z/t sont elles-mêmes irréductibles.

5.5.4 Ensemble $\mathbb{R}$ des réels

Les suites à valeurs dans $\mathbb{Q}$ n'ont pas forcément leur limite dans $\mathbb{Q}$!

---

[5] « / » sera considéré comme un *constructeur* de fraction ; par contre, « : » désignera la division entière comme opération ; ainsi, numérateur(6/3) = 6 et dénominateur(6/3) = 3, tandis que 6 : 3 = 2.

En effet, posons $x_0 = 1$, et $x_n = ½ (x_{n-1} + 2/x_{n-1})$; x_0 est rationnel, et x_n est rationnel dès que x_{n-1} l'est ; donc tous les x_n sont rationnels. Or $\lim_{n\to\infty} (x_n) = \sqrt{2}$ ne l'est pas, comme l'atteste une démonstration bimillénaire.

On a décidé d'appeler $\mathbb{R}$ l'ensemble des éléments de $\mathbb{Q}$ *et des limites des suites constructibles dans* $\mathbb{Q}$. $\mathbb{R} - \mathbb{Q}$ contient essentiellement des nombres dits *algébriques* et d'autres dits *transcendants*.

On montre qu'il existe toujours un élément de $\mathbb{Q}$ aussi proche que l'on veut de tout élément propre à $\mathbb{R}$ ([6]), et ceci donnerait bon espoir pour une arithmétique réelle calculable.

Malheureusement, on a aussi montré que dans [0 1] les éléments de $\mathbb{Q}$ sont dénombrables, tandis que ceux de $\mathbb{R}$ ne le sont pas (« puissance du continu »). De ce fait, les arithmétiques réelles peuvent être plus ou moins bien approximées par des arithmétiques rationnelles spécialisées, mais il n'existera pas en informatique de représentation parfaite des réels (qui ne seraient pas toujours finies) ni donc d'arithmétique parfaite pour les réels[7].

5.6 EXERCICES

5.6.0

Tout entier naturel est premier s'il n'est divisible par aucun entier naturel strictement compris entre 1 (exclus) et lui-même (exclus).

Etablir :

1. Au-delà de 2, tout nombre premier est impair.
2. Tout nombre impair est premier s'il n'est divisible par aucun nombre impair strictement compris entre 2 et lui-même.
3. Tout entier naturel est premier s'il n'est divisible par aucun nombre premier compris entre 1 (exclus) et lui-même (exclus).
4. Tout entier naturel est premier s'il n'est divisible par aucun entier naturel compris entre 1 (exclus) et sa racine carrée (incluse).
5. Tout entier naturel est premier s'il n'est divisible par aucun entier naturel premier compris entre 1 (exclus) et sa racine carrée (incluse).
6. Au-delà de 2, tout nombre premier est un nombre impair qui n'est divisible par aucun nombre impair compris entre 3 (inclus) et sa racine carrée(incluse).

On se demande si 959 est premier. D'après chacun des énoncés, combien doit-on envisager de faire de tests ?

5.6.1

Expliciter le rôle de la fonction quot : $\mathbb{N} \otimes \mathbb{N} \to \mathbb{N}$ définie par

$$\text{quot}(n, d) \equiv \{n<d \to 0 ; 1 + \text{quot}(n - d, d)\}.$$

5.6.2

Expliciter le rôle de la fonction mod : $\mathbb{N} \otimes \mathbb{N} \to \mathbb{N}$ définie par

$$\text{mod}(n, d) \equiv \{n<d \to n ; \text{mod}(n - d, d)\}.$$

---

[6] Techniquement, $\mathbb{Q}$ est dit *dense dans* $\mathbb{R}$.

[7] Les *virgules flottantes* et les *arithmétiques rationnelles* constituent en général les approximations rationnelles les plus heureuses qu'on sache faire en machine d'arithmétiques réelles qui restent impossibles.

5.6.3

Expliciter le rôle de la fonction pgcd : $\mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$ définie par

$$\text{pgcd}(p, q) \equiv \{p = q \rightarrow p \; ; \; p<q \rightarrow \text{pgcd}(q - p, p) \; ; \; \text{pgcd}(p - q, q)\}.$$

5.6.4

Expliciter le rôle de la fonction pgcd : $\mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$ définie par

$$\text{pgcd}(p, q) \equiv \{p \bmod q = 0 \rightarrow q \; ; \; q \bmod p = 0 \rightarrow p \; ; \; \text{pgcd}(p \bmod q, q \bmod p)\}.$$

5.6.5

Expliciter le rôle de la fonction YiKing : $\mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$ définie par

$\text{YiKing}(k, n) \equiv \{k = 0 \rightarrow 1 \; ; \; k = n \rightarrow 1 \; ; \; \text{YiKing}(k - 1, n - 1) + \text{YiKing}(k - 1, n)\}$.

5.6.6

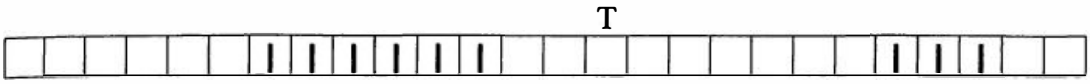
Expliciter le rôle de la fonction Legendre : $\mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$ définie par

$$\text{Legendre}(n,k) \equiv \{k = 0 \rightarrow 1 \; ; \; n = 0 \rightarrow 0 \; ; \; k \bmod 2 = 0 \rightarrow \text{Legendre}(n \cdot n, k : 2) \; ; \; n \cdot \text{Legendre}(n, k - 1)\}$$

Définir une version étendue Legendre' : $\mathbb{R} \otimes \mathbb{Z} \rightarrow \mathbb{R}$.

5.6.7

On considère une machine de Turing, dont le ruban ne comporte par case qu'un blanc ou un bâtonnet | .



- On code un entier n avec n + 1 bâtonnets consécutifs ; pourquoi ?
- On suppose la tête de lecture en T, dans l'état 0 ; montrer qu'on peut effectuer une *addition* en adjoignant successivement au second groupe chacun des bâtonnets du premier groupe (sauf le dernier : la machine doit l'effacer puis s'arrêter) ; définir les états et les règles nécessaires à cette opération.
- On suppose la tête de lecture en T ; montrer qu'on peut effectuer une *multiplication* du premier nombre par le second.
- Serait-ce plus simple en binaire ? en décimal ?
- Une élévation au carré doit-elle utiliser la multiplication ?

5.7 PROBLÈMES

5.7.0

Expliciter le rôle des fonctions : $\mathbb{N} \rightarrow \mathbb{B}$ définies par

estPremier(n) ≡ inDivisibilite(n, 2) ;

$$\text{inDivisibilite}(n, p) \equiv \{ (n : p) < p \rightarrow 1 \; ; \; (n \bmod p) = 0 \rightarrow 0 \; ; \; \text{inDivisibilite}(n, p + 1 + (p \bmod 2))\}.$$

Pourquoi appelle-t-on parfois *prédicat* ce genre de fonctions ?

5.7.1

Expliciter le rôle des fonctions : $\mathbb{N} \rightarrow \mathbb{N}$ définies par :

$\text{premier}(n) \equiv \{n<4 \rightarrow n \; ; \; \text{premier_a_partir_de}(\text{premier}(n - 1) + 2\}$

$$\text{premier_a_partir_de}(n) \equiv \{ \; \text{estPremier}(n) \rightarrow n \; ; \; (n \bmod 2) = 0 \rightarrow \text{premier_a_partir_de}(n + 1) \; ; \; \text{premier_a_partir_de}(n + 2)\}.$$

Peut-on avoir premier(n) = n + 10 ?

5.7.2

En supposant que dans $\mathbb{N}^{\oplus}$, ensemble des listes non vides d'entiers on a : <f> & <g, h> ≡ <f, g, h>, expliciter le comportement des fonctions à valeur dans $\mathbb{N}^{\oplus}$ qui sont définies par :

décomposition(n) ≡ liste_facteurs(n, 2) ;
liste_facteurs(n, f) ≡ { n < f·f → < n > ;
(n mod f) = 0 → < f > & liste_facteurs(n : f, f) ;
liste_facteurs(n, f + 1)
}

Etablir les valeurs de décomposition (11), décomposition (12), décomposition (92), décomposition (120).

5.7.3

On sait que red*((x/y)) ≡ (x : pgcd(x,y) / y : pgcd(x,y)).

Vérifier, sur des fractions telles que 36/1024, ou 27720/210 si on obtient le même résultat avec le système :

reduite (x, y) ≡ reduire(x, y, 2) ;
reduire(p, q, f) ≡ { p<f → (p/q) ;
q<f → (p/q) ;
(p mod f) = 0 ∧ (q mod f) = 0 → reduire(p : f, q : f, f) ;
reduire(p, q, f + 1)
}

5.7.4

Vérifier la convergence de la fonction $f_{91}:\mathbb{Z}\rightarrow\mathbb{N}$ définie par :

$f_{91}(n) \equiv \{ n < 0 \rightarrow f_{91}(n + 17) ; n > 3 \rightarrow f_{91}(n - 5) ; 91\}$.

5.8 PISTES DE RÉFLEXION

5.8.0

Arithmétique des polynômes à coefficients entiers.

5.8.1

Arithmétique en base B, où un (grand) entier n est exprimé par une suite d'entiers bornés par B, jouant le rôle de chiffres dans la base B. (ex : B = 60, B = 1000, B = 65 536...).

5.8.2

Repérer des « images récursives » dans la publicité, à la télévision. Connexion avec les fractals ? Connexion avec les L-systèmes[8] ? Application à la représentation des plantes, des coquillages...(cf. PRUSINKIEWICZ).

5.8.3

Récursivité et « mise en abyme » dans Gide et le Nouveau Roman.

5.8.4

Récursivité et structures hypotaxiques en linguistique (cf. TESNIÈRE, *Eléments de Syntaxe Structurale,* Editions Klincksieck, Paris, 1959-1988).

---

[8] Systèmes dynamiques modélisés par des grammaires, proposées dès 1968 par Aristid Lindenmayer.

5.9 LECTURES

BENZAKEN C., *Système formels, introduction à la logique et à la théorie des langages,* chap. 5, Masson, 1991.

GROSS M., LENTIN A., *Notions sur les grammaires formelles,* chap. 4 et 5, Gauthier-Villars, 1967.

PRUSINKIEWICZ P., HANAN J., *Lindenmayer Systems, Fractals, and Plants,* coll. Lecture notes in biomathematics, vol. 79, Springer Verlag 1989/92 (et la suite des *The algorithmic beauty of* ...).

REINHARDT F. & SOEDER H., *Atlas des mathématiques,* chap. 4, La Pochothèque, Librairie Générale Française, 1997.

VÉLU J., *Méthodes mathématiques pour l'informatique,* chap. xiv, Dunod, 1994.

VOROBIEV N., *Caractères de divisibilité ; suite de Fibonacci*, Collection « Initiations aux Mathématiques », Editions de Moscou, 1973.

Collectif, *Petite encyclopédie des mathématiques,* sections 1.4, 14.5, 15.4, 31, Edition Pagoulatos, Didier distr., 1980.

Voir aussi des sites comme http://plato.stanford.edu/entries/church-turing.

CHAPITRE 6

NOTION DE COMPLEXITÉ

En amont de la programmation, on a voulu se doter d'une mesure de complexité pour prévoir l'ordre de grandeur des ressources que l'exécution d'un algorithme consommerait.

D'un point de vue informatique, il s'agira :

- en matière de développement, de caractériser une solution avant de la programmer ; en effet, il est peut être plus urgent de reprendre l'analyse, si la solution s'avère déjà trop vorace ;
- dans le cadre d'une optimisation ou d'une rétro-conception, d'apprécier l'origine de la voracité excessive de telle ou telle partie d'une application existante.

Nous abordons le sujet à travers l'analyse de la complexité de fonctions.

D'un point de vue mathématique, on adjoindra ainsi au classique critère d'*élégance* – critère statique généralement lié à la concision du texte – un critère d'*efficience* du processus de calcul correspondant.

6.1 PREMIER EXEMPLE

Comparons deux procédés d'*élévation à la puissance entière p d'un nombre n*.

Considérons la fonction $e(n, p) \equiv \{n = 0 \rightarrow 0 ; p = 0 \rightarrow 1 ; e(n, p-1)\cdot n\}$.

Par récurrence sur p, on trouve $e(n, p) = n^p$.

Appelons t(p) le temps nécessaire pour calculer e(n, p) avec un n non nul. Nous aurons en général $t(p) = t(p-1) + k$, où la constante k représente le temps nécessaire pour « rater » les deux premiers tests et faire la multiplication, et nous pouvons de même poser $t(0) = m$. D'où, par récurrence, une *loi* $t(p) = k\cdot p + m$, qui indique que *le temps d'exécution sera une fonction linéaire de p*.

Considérons maintenant la fonction

$$\begin{aligned} \text{Legendre}(n, p) \equiv \{ & n = 0 \rightarrow 0 ; p = 0 \rightarrow 1 ; \\ & p \bmod 2 = 0 \rightarrow \text{Legendre}(n\cdot n, p : 2) ; \\ & n\cdot\text{Legendre}(n, p-1) \} \end{aligned}$$

qui satisfait aussi $\text{Legendre}(n, p) = n^p$. La durée t(p) correspondante est cette fois donnée par :

$$t'(p) = \{p = 0 \rightarrow m' ; p \bmod 2 = 0 \rightarrow t(p : 2) + k' ; t(p-1) + k''\}$$

En désignant par *ld* [1] le logarithme à base 2, et par $\lfloor\ \rfloor$ la fonction plancher (voir § 3.2.1), on peut encadrer t′ (p) :

$$k'' + k'\lfloor \mathrm{ld}(p) \rfloor \leq t'(p) \leq k' + (k' + k'')\lfloor \mathrm{ld}(p) \rfloor$$

---

[1] ld pour *logarithmus dualis*.

La borne inférieure est atteinte pour les $p = 2^k$ et la borne supérieure pour les $p = 2^k - 1$ soit {3, 7, 15...}.

La première fonction est la plus concise, mais cette simplicité se révèle onéreuse. Aux faibles valeurs de p, il est difficile de dire quelle est la fonction dont l'évaluation est la plus rapide. Par contre, aux valeurs « suffisamment élevées », nous pouvons recourir à la notation de Bachmann (1895) et écrire :

$$t(p) = O(p), \text{ et } t'(p) = O(\,\mathrm{ld}(p)\,)$$

pour dire que t a un ***comportement asymptotique*** linéaire, et t' un *comportement asymptotique* logarithmique.

Aussi, dès que p sera « suffisamment élevé », la seconde fonction calculera les n^p plus vite que la première.

Si maintenant on s'intéresse non au temps mais au nombre de multiplications, il n'y a plus lieu de distinguer les k, k', k''... On aura $m(p) = p$ et $1 + \lfloor \mathrm{ld}(p) \rfloor \leq m'(p) \leq 1 + 2\lfloor \mathrm{ld}(p) \rfloor$.

Tableau 6.1 Complexité de n^p.

m(p) = p	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Borne inférieure m'		*1*	*2*	2	*3*	3	3	3	*4*	4	4	4	4	4
m'(p)	0	*1*	*2*	*3*	*3*	4	4	*5*	*4*	5	5	6	5	6
Borne supérieure m'		*1*	3	*3*	5	5	5	*5*	7	7	7	7	7	7

Adaptons nos fonctions à la définition des puissances entières de nombres réels, les opérations étant faites avec une précision relative ε. La fonction e calculera une puissance n-ième avec n multiplications, soit une incertitude relative nε sur le résultat, tandis que Legendre entraînera une erreur relative de l'ordre de 2.ld(n).ε au pire. Pour une puissance $100^{\text{ième}}$, la première solution mènerait à une incertitude de 100ε, contre 12ε pour la seconde. L'avantage de la seconde solution devient écrasant pour les puissances 1000 et plus, à supposer qu'on les emploie !

Cet exemple montre bien :

- qu'on doit pouvoir définir diverses mesures de complexité ;
- que celles-ci donneront souvent des prédictions intéressants, dès lors qu'on raisonne sur des valeurs suffisamment grandes des arguments : *le comportement asymptotique s'attache aux tendances lourdes extrapolables plutôt qu'aux idiosyncrasies des cas élémentaires ;* en effet, ces cas sont à la fois négligeables car généralement peu coûteux, et d'apparence irrégulière car les initialisations s'y dégagent mal des traitements généraux qui dictent les lois tendancielles.

6.2 RÈGLES D'ESTIMATION DE LA COMPLEXITÉ

6.2.1 Indices de complexité

Prédire les comportements se heurtera au fait qu'en l'absence de certaines données on ne pourra pas toujours *évaluer* mais seulement *encadrer* la complexité. Il pourrait alors être intéressant de distinguer des complexités au mieux et au pire, éventuellement notées comme un intervalle. Avec des données probabilistes sur les

données (nombre, distribution), on peut aussi s'intéresser à une complexité « moyenne ».

6.2.2 Evaluation de la durée

On note t(p) la complexité en temps du processus p.

Opérations de base

Il s'agira d'opérateurs, d'affectations, d'entrées/sorties considérées comme atomiques. Leur complexité sera considérée comme constante. On utilisera soit *0* pour les opérations négligées/négligeables soit *1* pour une opération normale soit une *constante symbolique* pour les opérations dont on veut étudier l'importance relative.

Si accéder à un élément de rangée (tableau à 1 dimension) doit être considéré comme une opération, on notera que t[i, j, k] est une sténographie de ((t[i])[j])[k], et qu'ainsi un accès à une table à n dimensions coûte n fois l'accès à une rangée, du fait du calcul d'adressage sous-jacent.

Invocation de fonction

L'évaluation des arguments est éventuellement distinguée de l'évaluation de la séquence ou du bloc formant le corps de la fonction

$$t(f(expr1, expr2, expr3)) = t(expr1) + t(expr2) + t(expr3) + t(f(x,y,z)\ \textit{pour}\ x = val(expr1), y = val(expr2), z = val(expr3))$$

Séquences et blocs

t({i1 ; i2 ; i3}) = t(i1) + t(i2) + t(i3) s'il s'agit uniquement d'une séquence d'instructions.

t({*local var1, var2* ; séquence}) = 2·t(varloc) + t(séquence) pour un bloc, si on veut expliciter la gestion (création + libération) de 2 variables locales.

Section parallèle

Au mieux :

$$t(p1 \mathbin{//} p2 \mathbin{//} p3) = \max(t(p1), t(p2), t(p3)),$$

à pénaliser éventuellement d'une durée d'installation ou de lancement.

Conditionnelles

$$t(\{\textit{si}\ cond\ \textit{alors}\ b0\ \textit{sinon}\ b1\}) = t(cond) + (\textit{si}\ cond\ \textit{alors}\ t(b0)\ \textit{sinon}\ t(b1)) \leq t(cond) + \max(t(b0), t(b1)).$$

$$\begin{aligned} & t(\{c0 \rightarrow b0 ; c1 \rightarrow b1 ; c2 \rightarrow b2 ; b3\}) \\ = \quad & t(c0) + (c0 \rightarrow t(b0) ; t(c1) + \{c1 \rightarrow t(b1) ; t(c2) + t(\{c2 \rightarrow t(b2) ; b3\})\}) \\ \leq \quad & t(c0) + \max(t(b0), t(c1) + \max(t(b1), t(c2) + \max(t(b2) + t(b3)))). \end{aligned}$$

Si toutes les conditions se ramènent aux différentes valeurs d'un même sélecteur à valeur dans un ensemble fini, la durée de sélection peut être unique (sélection *en rateau* ramenée à un aiguillage par table de branchement)

$$t(\{\textit{selon}\ i : 0 \rightarrow b0 \mid 1 \rightarrow b1 \mid 2 \rightarrow \ldots \mid 77 \rightarrow b77 \ldots \mid b999\}) = t(sel(i)) + \max(t(b0), t(b1), \ldots, t(b77), \ldots, t(b999))$$

Itérations

t({*faire* n *fois* séquence}) = n · t(séquence)

t({*pour i = k à n faire* séquence(i)}) = $\Sigma^{n}_{i=k}$ (t(pour) + t(séquence(i)))

$\leq (n - k + 1)\cdot(t(pour) + \max_i(t(séquence(i)))$

t({*tantque* cond *faire* séquence}) =

t(cond) + (cond → t({séquence ; *tantque* cond *faire* séquence}) ; 0)

De toutes façons, il faut évaluer la condition ; si elle est vraie, on exécute la séquence, et on revient à la boucle dans un contexte renouvelé.

Récursivités

L'évaluation de la durée mène en général à une récurrence calquant la récursivité de la définition, le coût d'un appel de fonction étant souvent considéré comme primordial[2].

6.2.3 Besoins en mémoire

On notera m(p) l'évaluation des besoins en mémoire d'un processus p, souvent composites.

Les besoins en *mémoire statique* se combinent additivement[3] ; toutefois, les implantations en recouvrement se combinent par max.

Les besoins en *mémoire dynamique* ont pour valeur la plus grande somme des besoins relatifs aux blocs et appels de fonction simultanément actifs ; ainsi,

- les demandes du bloc B1 s'ajoutent à celle de B0 si B1 est inclus dans B0 ;
- si B0 et B1 sont disjoints, m({B0 ; B1}) = max(m(B0), m(B1)).

De même, si la fonction f appelle la fonction g, leurs besoins (au sens de leurs articles d'activations et de leurs blocs propres) s'ajoutent ; par contre, si on appelle f, puis g quand f est terminée, leurs besoins respectifs se combinent par max.

Pour les fonctions récursives, on doit faire la même analyse en recherchant le cas de la plus forte demande.

Pour les récursivités directes (f(a) appelle f(b)...) cette demande est généralement fixée par la ***profondeur de récursivité*** (nombre maximum d'appels simultanément en cours d'une même fonction).

Pour les récursivités croisées ou indirectes, on procédera de même pas à pas : si f(a) appelle g(a) qui appelle h(b) qui appelle f(c), on explicitera les besoins globaux de f(a) en cumulant les besoins locaux de f(a), g(a), h(b) aux besoins globaux de f(c).

6.2.4 Autres cas

On adapte les raisonnements ci-dessus – mutatis mutandis.

---

[2] Pour une fonction à k paramètres et n variables locales, un appel à cette fonction implique physiquement la gestion d'un article d'activation, créé à l'appel et libéré au retour, comportant k emplacements pour les paramètres, initialisés par l'évaluation des k arguments, plus n emplacements pour les variables locales propres à la fonction.

[3] Attention : une donnée statique n'est pas toujours visible.

6.3 PRÉORDRE DE COMPLEXITÉ

DÉFINITION 6.1 – Soit f et g deux fonctions à 1 argument. On dira f ***pas plus complexe que*** g si on a :

$$f = O(g) \equiv \lim_{n\to\infty} f(n) / g(n) = k$$

où k est une constante réelle finie.

EXEMPLES

- On a toujours $f = O(f)$; si k' est une constante finie non nulle : $k'\cdot f = O(f)$; si de plus f est asymptotiquement croissante $f + k' = O(f)$.
- $5n + 12 = O(5n)$, $5n = O(n)$ et $5n + 12 = O(n)$; par ailleurs $8n - 7 = O(n)$; on n'a pas en général $5n + 12 = 8n - 7$, mais seulement $5n + 12 = O(8n - 7)$ et $8n - 7 = O(5n + 12)$. Il ne faut donc pas considérer O comme un opérateur, mais *considérer = O comme une relation entre fonctions.*
- $f + g = O(\max(f,g))$, et notamment $x^m + x^n = O(x^{\max(m,n)})$.
- Finalement, si $P_n(x)$ est un polynôme de degré n, $P_n(x) = O(x^n)$.

PROPOSITION 6.1 – « = O » est un *préordre* sur $\mathbb{N}^{\mathbb{N}}$.

PREUVE

- $\forall f, f = O(f)$ /réflexivité
- $\forall f, g, h\ f = O(f)$ et $g = O(h) \Rightarrow f = O(h)$ /transitivité ■

DÉFINITION 6.2 – On dira f et g de ***complexité similaire*** si on a :

$$f = \Theta\, g \equiv f = O(g) \wedge g = O(f) \equiv \lim_{n\to\infty} f(n) / g(n) = k,$$

avec k, constante *non nulle.*

PROPOSITION 6.2 – « = Θ » est l'équivalence standard attachée au préordre « = O ».

DÉFINITION 6.3 – On dira f ***moins complexe que*** g si on a :

$$f = o(g) \equiv f = O(g) \wedge \neg (g = O(f)) \equiv \lim_{n\to\infty} f(n) / g(n) = 0$$

PROPOSITION 6.3 – « = o » est le préordre strict attaché au préordre large « = O ».

6.4 CLASSES DE COMPLEXITÉ

6.4.1 Classes élémentaires

Θ introduit un système de classes d'équivalence entre fonctions, classes liées entre elles par un ordre de complexité.

Voyons les classes les plus courantes. On désigne chacune par un représentant,

1	pour la classe des	constantes
log(x)		formes logarithmes
x		formes linéaires en x, ou de degré 1
x^n		polynômes de degré n en x
		(resp. ***posynômes*** si n est rationnel non entier)
k^x		formes exponentielles de base k.

Ces classes seront ordonnées ainsi : classe 1, classe log, polynômes et *posynômes* dans l'ordre croissant de leurs degrés, classes exponentielles dans l'ordre de leurs bases..., puis les exponentielles de polynômes...

REMARQUE. Cette classification est basée sur le *comportement asymptotique des rapports de complexité.* Ainsi, un comportement en $375x^7$ (de la classe x^7) sera dit strictement *moins complexe* qu'un comportement en $0{,}000001\ x^8$ (de la classe x^8), et l'on notera, s'il y a lieu, $375x^7 = o(0{,}000001x^8)$, car $x^7 = o(x^8)$. *Les apparences contraires voudront simplement dire que les valeurs de x « suffisamment grandes» pour que ce soit vrai, devront être particulièrement élevées (par exemple $x > 10^9$).*

6.4.2 Utilisation

Cette classification permet d'évaluer simplement certains cas de figure ; par exemple, en matière de temps d'exécution,

- les durées logarithmiques ou linéaires sont recherchées,
- les durées polynomiales plus ou moins acceptables,
- les durées « exponentielles » considérées comme non praticables.

Cela facilite la comparaison de diverses heuristiques s'adressant au même problème, pour choisir celle à adopter ou celles méritant d'être testées.

Avec une approche multicritère utilisant diverses mesures de complexité, relatives à la durée, la mémoire centrale nécessaires, l'ampleur des fichiers temporaires, et aussi des mesures de qualité telles que la précision, le choix ne sera peut être pas totalement tranché, mais permettra au moins d'*éliminer les méthodes les moins défendables*, au profit de celles ayant des avantages marqués.

6.4.3 Autres cas

On peut tomber sur des cas renvoyant à des classes moins / peu connues, et dont on ne connaît pas le classement a priori. Les raisonnements classiques sur les comportements asymptotiques s'appliquent.

EXEMPLES

- Le rectorat demande de trier d'urgence les dossiers de myriades de candidats bacheliers ; vaut-il mieux le tri A, en $n \cdot ld(n)$, ou le tri B, en $n^{1,235}$? Le problème revient à comparer $ld(n)$ à $n^{0,235}$. Classiquement, on considère que toute fonction logarithmique de n croit (asymptotiquement) moins vite que toute puissance positive de n. Donc le tri A devrait être moins monstrueux que le tri B (et si ce n'est pas encore vrai pour 60.000, ça le sera pour 300.000 !).

- Mon problème de combinatoire me ramène à 3 méthodes, la méthode A en n!, la méthode B en n^n, la méthode B en 2^{n^2}. Avec laquelle continuer ?
  - $n! \approx (n/e)^n (2\pi n)^{1/2}$, par suite $(n!/n^n) \approx e^{-n} (2\pi n)^{1/2} \to 0$ quand n croît, et $n! = o(n^n)$: A est strictement moins complexe que B ;
  - $ld(n^n) = n \cdot ld(n)$, et $ld(2^{n^2}) = n^2$; $ld(n) = o(n)$, $n \cdot ld(n) = o(n^2)$ et a fortiori $n^n = o(2^{n^2})$. B est significativement moins complexe que C. *Je prends donc A.*

Quand n croît, A est *asymptotiquement* le moins vorace. Et pour n = 10, je « choisis » une complexité de $3{,}6 \cdot 10^6$ contre 10^{10} (B) et $2^{100} = 1{,}27 \cdot 10^{30}$ (C) ; mais ces nombres sont déjà énormes, et je dois rester prudent : pour n = 20 puis n = 30 la situation s'aggrave rapidement, car

1 jour = $10^{4,94}$ secondes fait que

1 an = $10^{2,56}$ jours = $10^{7,5}$ secondes = $10^{13,5}$ microsecondes

et l'âge de la Terre est sensiblement de 10^{23} microsecondes

La méthode A, *la moins pire*, est vite inabordable, et il faudrait trouver mieux...

Tableau 6.2 Formes asymptotiques et ordres de grandeur.

n	n^2	n^3	2^n	exp(n)	n !	n^n	2^{n^2}	2^{2^n}
5	25	125	32	148	120	3125	$3{,}35\cdot10^6$	$4{,}3\cdot10^9$
10	100	1000	1024	22026	$3{,}6\cdot10^6$	10^{10}	$1{,}27\cdot10^{30}$	10^{308}
20	400	8000	$1{,}05\cdot10^6$	$4{,}85\cdot10^8$	$2{,}4\cdot10^{18}$	$1{,}05\cdot10^{26}$	$2{,}6\cdot10^{120}$	$10^{1.05X\,XXX}$
30	900	27000	$1{,}07\cdot10^9$	$1{,}07\cdot10^{13}$	$2{,}65\cdot10^{32}$	$2{,}05\cdot10^{44}$	$2{,}5\cdot10^{270}$	$10^{1.07X\,XXX.XXX}$

6.5 OPTIMISATION DU CALCUL D'UNE FONCTION

Au XIIIe siècle, Leonardo Pisano, à propos de l'évolution d'une population de lapins, introduisit une suite 1, 1, 2, 3, 5, 8..., de nombres dits depuis « de Fibonacci », qui, au-delà des valeurs initiales, forme chaque nouveau terme comme somme des deux précédents (cf. *Petite encyclopédie des mathématiques*, VOROBIEV et autres sources). En introduisant une valeur 0 au rang 0, on définit maintenant les termes de cette suite à l'aide de la fonction fib :

$$\text{fib}(n) \equiv \{n = 0 \rightarrow 0 \,;\, n = 1 \rightarrow 1 \,;\, \text{fib}(n-1) + \text{fib}(n-2)\}$$

Une telle définition est *calculable,* mais est-elle *opérationnelle* ?

6.5.1 Comportement de la fonction

Le calcul montre que cette fonction est rapidement croissante, et l'on soupçonne un comportement exponentiel.

Pour a, k, b convenables, a-t-on $\text{fib}(n) \equiv a\cdot k^n + b$?

La récurrence donnerait : $\text{fib}(i) = \text{fib}(i-1) + \text{fib}(i-2)$ soit :
$$a\cdot k^n + b \equiv a\cdot k^{n-1} + b + a\cdot k^{n-2} + b$$

En identifiant terme à terme[4] il vient :

- $b = 2b$ soit $b = 0$ (6.1)
- $a\cdot k^n = a\cdot k^{n-1} + a\cdot k^{n-2}$ (6.2)
- $k^2 = k + 1$, en simplifiant par $a\cdot k^{n-2}$. (6.3)

(6.3) admet comme racines $k = (\sqrt{5} \pm 1)/2$; la récurrence ayant visiblement comme solution toute combinaison linéaire de ses solutions de base, il y a lieu de supposer que :

$\text{fib}(n) = a_1\cdot k_1^n + a_2\cdot k_2^n$, où k_1 et k_2 sont les deux solutions trouvées pour k.

Les valeurs initiales imposent finalement :

$$\text{fib}(n) = \{(\sqrt{5}+1)/2\}^n - \{(\sqrt{5}-1)/2\}^n$$

et le joli paradoxe d'une fonction entière différence de deux termes irrationnels.

Le second terme est rapidement négligeable, et $\text{fib}(n) \approx 1{,}618^n$, soit grosso modo $e^{0{,}481n}$.

[4] La réussite de l'identification terme à terme prouvera le bien-fondé de l'hypothèse.

6.5.2 Complexité / durée de calcul

Que coûterait en temps l'évaluation directe de

$$\text{fib}(n) = \{n = 0 \rightarrow 0 \; ; \; n = 1 \rightarrow 1 \; ; \; \text{fib}(n-1) + \text{fib}(n-2)\} \; ?$$

Supposons que la durée soit essentiellement due aux appels de fonction ; leur nombre a(n) est régi par :

$$a(n) = \{n = 0 \rightarrow 1 \; ; \; n = 1 \rightarrow 1 \; ; \; a(n-1) + a(n-2) + 1\}.$$

Pour éliminer le terme constant de la récurrence, on pose a(n) = b(n) + k d'où

$$b(n) + k \equiv b(n-1) + k + b(n-2) + k + 1.$$

k s'élimine pour $k \equiv 2k + 1$, satisfait pour $k = -1$, d'où

- $a(n) = b(n) - 1$ (6.4)
- $b(n) = b(n-1) + b(n-2)$ (6.5)

La suite a = {1, 1, 3, 5, 9, 15, 25...} renvoie à une suite b = {2, 2, 6, 10, 16, 26...}, du type fib(n) d'après la récurrence (6.5). Effectivement, $b(n) = 2 \cdot \text{fib}(n+1)$ d'où

$$a(n) = 2 \cdot \text{fib}(n+1) - 1 \approx 3{,}23 \; \text{fib}(n) \text{ ou } 1{,}62 \cdot a(n-1)$$

et la *durée de calcul, proportionnelle au résultat, est une fonction exponentielle de l'argument,* ce qui doit la faire considérer comme désastreuse[5].

Cependant, cette valeur se comprend bien, si on voit l'évaluation comme le développement d'une arborescence (fig. 6.1)

- dont chaque nœud, y compris la racine, serait porteur d'un « + »,
- dont chaque feuille porterait soit un 0 soit un 1.

Le nombre de feuilles à 1 est précisément le résultat, le nombre de feuilles à 0 et le nombre de nœuds étant similaires...

6.5.3 Complexité / taille mémoire

L'évaluation étudiée est-elle aussi vorace en mémoire ?

Techniquement, quelle est la ***profondeur maximale de la récursivité*** (i.e. le nombre de fib() simultanément en cours d'évaluation) ?

La réponse nous est donnée par la *longueur maximale d'un chemin actif dans l'arbre d'évaluation* évoqué. Au pire, fib(n) commence à évaluer fib(n – 1), et à son sujet fib(n – 2), fib(n – 3) ... et enfin fib(1), soit au plus n *évaluations simultanées.*

La mémoire nécessaire pour évaluer ainsi fib(n) varie au plus en n (fig 6.1).

---

[5] C'est pourquoi le calcul de fib(20) selon cette formulation fut longtemps l'étalon de vitesse, facilement observable, des langages comme *lisp* ou *scheme.*

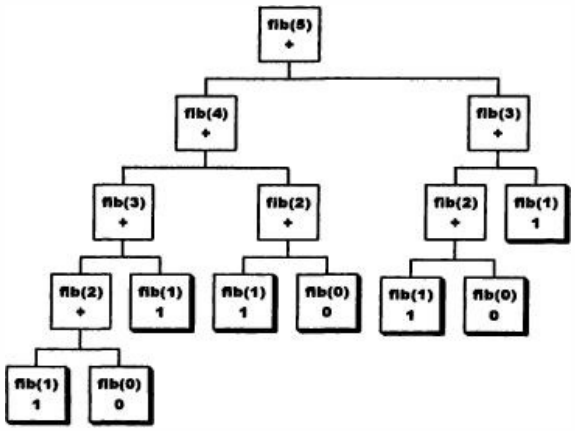


Fig. 6.1 Arbre d'évaluation de fib(5).

6.5.4 Premières variantes

En quête d'accélération, nous notons que fib(n) = fib(n – 1) + fib(n – 2) fait qu'en général fib(n – 1) = fib(n – 2) + fib(n – 3), d'où en remplaçant fib(n – 1), la formule de récurrence additive :

$$\text{fib}(n) = 2\cdot\text{fib}(n-2) + \text{fib}(n-3)$$

en remplaçant fib(n – 2), la formule de récurrence soustractive :

$$\text{fib}(n) = 2\cdot\text{fib}(n-1) - \text{fib}(n-3)$$

Le procédé est itérable, si bien que nous pouvons envisager le cas fib(n) = s(fib(n – k), fib(n – m)) où fib(n) est *somme pondérée* de fib(n – k) et fib(n – m), avec k<m<n.

La formulation

$$\text{fib}(n) \equiv \{n = 0 \rightarrow 0 \,;\, n = 1 \rightarrow 1 \,;\, 2\cdot\text{fib}(n-2) + \text{fib}(n-3)\}$$

serait divergente ; nous retiendrons donc une formulation modifiée, convergente sur $\mathbb{N}$:

$$\text{fib}(n) \equiv \{n<3 \rightarrow \{n = 0 \rightarrow 0 \,;\, 1\} \,;\, 2\cdot\text{fib}(n-2) + \text{fib}(n-3)\}$$

De façon générale, pour une récurrence fib(n) = s(fib(n – k),fib(n – m)), la plage d'arrêt devrait comprendre *au moins m valeurs initiales consécutives.*

Supposons la somme pondérée de coût négligeable devant les appels de fonction. Le schéma général :

$$\text{fib}(n) \equiv \{\ \text{initial}(n) \rightarrow \text{valeurs prédéfinies} \,;\, s(f(n-k), f(n-m))\}$$

mène à un nombre d'appels :

$$a(n) = \{\text{initial}(n) \rightarrow 1 \,;\, 1 + a(n-k) + a(n-m)\}$$

Posant $a(n) = b\cdot r^n + c$, on obtient la récurrence :

$$b\cdot r^n + c = (b\cdot r^{n-k} + c) + (b\cdot r^{n-m} + c) + 1,$$

dont c s'élimine facilement en prenant $c = 2\cdot c + 1$ soit $c = -1$.

Par exemple, l'expression	mène à	de racine dominante
$fib(n) = 2 \cdot fib(n-2) + fib(n-3)$	$r^3 = r^2 + 1$,	1,4656
$fib(n) = 2 \cdot fib(n-1) - fib(n-3)$	$r^3 = r + 1$,	1,3247
$fib(n) = 3 \cdot fib(n-3) + 2 \cdot fib(n-4)$	$r^4 = r + 1$,	1,2207.

La durée nécessaire, toujours à croissance exponentielle, est significativement plus courte, de la formulation initiale (r = 1,618) au schéma soustractif (r = 1,4656) puis aux deux schémas additifs.

Pour la mémoire, on raisonne comme précédemment : au pire, fib(n) appelle fib(n – k), fib(n – 2k) ... pour finir sur la plage d'arrêt : pour n donné, il y a donc jusqu'à $\lceil n/k \rceil$ évaluations simultanément en cours.

Ainsi, l'évaluation de la fonction fib fondée sur une récurrence

$$fib(n) = s(fib(n-k), fib(n-m)), \qquad \text{où } k<m,$$

- reste de durée exponentielle, mais s'accélère pour des schémas plus élaborés → rechercher k et m élevés
- suppose au moins m valeurs initiales → rechercher m (>k) faible
- occupe « en pointe » une mémoire en n/k. → rechercher k (<m) élevé

Les meilleures formulations seraient fournies avec la relation m = k + 1 des schémas additifs.

6.5.5 Linéarisation de la durée d'évaluation

Ce qui fait le caractère exponentiel des formulations de fib rencontrées jusqu'ici, c'est l'aspect arborescent (ou pyramidal) de l'évaluation, qui sans mémoire aucune réévalue un nombre croissant de fois les valeurs à petit argument.

On aurait une évaluation plus directe si, à tout moment, fib(i) et fib(i – 1) étaient simultanément disponibles. Soit donc fib2(n) une fonction capable de produire la *paire* <fib(n) fib(n – 1)>. Il suffira d'écrire par exemple :

fib(n) ≡ {premier(fib2(n))}
fib2(n) ≡ {n = 0 → <0, 1> ; successeur (fib2(n – 1))}

où *successeur* calcule <fib(n), fib(n–1)> à partir de <fib(n – 1), fib(n – 2)>, soit :

successeur(paire) ≡ {<premier(paire) + second(paire), premier(paire)>}.

Il est maintenant clair que le nombre d'appels pour fib et fib2 est donné par a(n) = n + 1, tandis que pour successeur : on a s(n) = n.

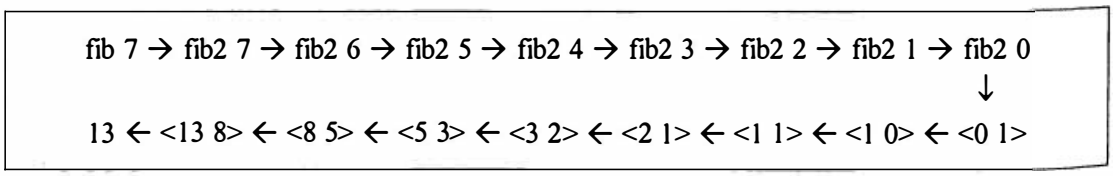


Fig. 6.2 Schéma linéaire d'évaluation de fib (7).

6.5.6 Usage de variable à affectation unique

Cette fonction *succcesseur*, très spécifique, n'est invoquée qu'une fois. Peut-on remplacer son invocation par son développement ? Si on veut tirer 2 usages de la paire fabriquée par fib2 sans invoquer 2 fois cette fonction, une affectation de cette paire à une *variable relais* est indispensable.

Introduisons le schéma {soit <var> = <expr> ; ... } pour signifier que dans la partie ... chaque occurrence de <var> a pour valeur <expr>, *sous-expression nommée pour ne l'évaluer qu'une fois.* Nous avons maintenant :

fib(n) ≡ {premier(fib2(n))}

fib2(n) ≡ { n = 0 → <0, 1> ;
{soit paire = fib2(n – 1) ;
<premier(paire) + second(paire), premier(paire)>}
}

« paire » est initialisée avec un signe =, car la valeur reçue à la création ne peut être modifiée : de ce fait, notre « variable locale » est plutôt une constante éphémère.

6.5.7 Passage à l'itération

La fonction fib2 ci-dessus relève du schéma

$$fib2(n) \equiv \{ n = 0 \rightarrow fib_0 ; g(fib2(n-1)) \}$$

- qui se ramène à $fib2(n) \equiv \{ g^n(fib_o) \}$;
- qu'on peut réaliser par une itération simple, à condition d'admettre des *affectations répétées* à une même variable simple « temp » :

$$fib2(n) \equiv \{ \text{soit temp} \leftarrow fib_0 ; \text{faire n fois } \{\text{temp} \leftarrow g(\text{temp})\} \}$$

En reprenant également l'invocation de fib2 par fib, on obtient finalement :

fib(n) ≡ { soit paire ← <0, 1> ;
faire n fois {paire ← <premier(paire) + second(paire), premier(paire)>} ;
premier(paire) }

Si cette définition est plus lourde que la première au niveau textuel, on notera :

- la *durée d'exécution en O(n),* faible car on n'utilise que des primitives,
- la *mémoire nécessaire, réduite à une variable locale.*

6.6 REPRÉSENTATIONS D'UNE TABLE CREUSE

On se pose ici la question de la représentation concrète d'une table abstraite, sous l'angle des complexités en taille et en temps d'accès.

6.6.1 Propriétés des tables

Soit à exploiter une table T formée de m lignes et n colonnes.
L'implantation directe d'une telle table

- a un encombrement en m·n,
- suppose, pour un T[i, j] quelconque, un temps d'accès constant, correspondant à la durée d'évaluation de l'adresse par une formule du genre a + i·b + j·c (où a est une adresse de base, et où b et c sont des déplacements convenables).

6.6.2 Représentation des tables creuses par une liste

Supposons la table très grande (par exemple 400·700), avec une faible proportion α des cases de la table occupées par une valeur significative – par exemple 1% – les autres cases contenant une même valeur connue – que nous dirons la *valeur banale*. L'idée est de ne représenter physiquement que les α·m·n valeurs remarquables.

Utilisons pour cela une table de listes (fig. 6.3) comprenant d'abord une table de m pointeurs de ligne, le i-ème pointeur

- renvoyant sur la liste représentant la i-ème ligne, si cette ligne contient au moins une valeur remarquable,
- étant vide si la ligne correspondante ne contient que la valeur banale.

Chaque liste représentant une ligne est formée de triplets (indice, valeur, suivant) contenant respectivement l'indice (colonne) de la case représentée, la valeur de la case, et un pointeur sur l'éventuelle cellule suivante.

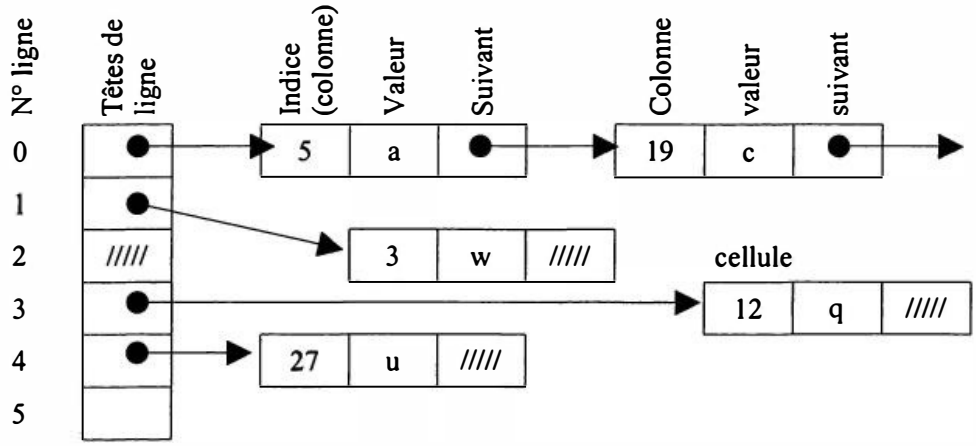


Fig. 6.3 Table creuse implantée en table de listes.

Encombrement

Prenons comme unité la place nécessaire pour une valeur, et posons que les pointeurs sont de taille p et les cellules de taille k > p. L'encombrement de la table de listes est alors p·m + α·m·n·k = m(p + α·n·k). En négligeant p devant α·n·k, on voit que l'*encombrement* est en O((α·k)·(m·n)).

Temps d'accès

Par ailleurs, le *temps d'accès* à une case [i, j] suppose d'abord un accès à la ligne i avec un temps constant faible (1 dimension), puis une migration dans la liste trouvée, pour trouver la case j cherchée ; présente, on accède directement à sa valeur ; absente, elle indique une valeur banale.

Avec une probabilité α qu'une case soit à représenter par une cellule,

- une case a une probabilité $(1 - \alpha)$ de ne pas l'être,
- une ligne de n cases une probabilité $(1 - \alpha)^n$ de ne pas être représentée par une liste.

Supposons $m(1 - \alpha)^n$ petit devant 1 : les m lignes donnent m listes effectives contenant en moyenne n·α cellules. La migration dans une telle liste se termine soit en atteignant la case cherchée, soit en constatant son absence, ce qui dénote la valeur

banale. La durée moyenne de migration dans une telle liste est en $n \cdot \alpha$ si elle n'est pas triée selon l'indice de colonne, en $n \cdot \alpha/2$ si elle est maintenue triée sur cet indice : le temps d'accès est donc en $O(n \cdot \alpha)$.

6.6.3 Discussion

La table 6.3 résume les 2 cas étudiés en termes de complexité, concernant la représentation physique d'une table abstraite à 2 dimensions.

Tableau 6.3 Représentation d'une table creuse à 2 dimensions.

	Encombrement	Durée d'accès
Table à 2 dimensions	$O(m \cdot n)$	$O(1)$
Table de listes	$O(\alpha \cdot k \cdot m \cdot n)$	$O(n \cdot \alpha)$

La table de listes est une bonne idée :

- si la table d'origine est creuse ($\alpha << 1$), et les cellules pas trop monstrueuses vis-à-vis du champ valeur ($k \cdot \alpha << 1$, condition physique) ;
- si le temps d'accès, de l'ordre de $n\alpha$ fois plus élevé, ne devient pas prohibitif.

Avec $m = 400$, $n = 700$, $\alpha = 1\%$,

- le gain en place est net (si la valeur prend 1 octet, l'indice 2, le pointeur 4, $k = 7$, d'où gain d'un facteur 15),
- le temps d'accès est multiplié par 7 environ.

Si $\alpha = 10\%$, le gain en place n'est important que si k est assez faible, la valeur occupant la plus grosse partie d'une cellule : complexe, chaîne, n-uple... ; tandis que le temps d'accès est environ 70 fois celui d'une table classique.

Une telle estimation préalable peut éclairer les choix de représentation, et l'algorithmie associée.

6.7 EXERCICES

6.7.0

Montrer que la somme ou le produit des éléments d'un vecteur de taille n a une durée en n, de même que la recherche de son max ou de son min.

6.7.1

On dispose d'une machine sur laquelle plusieurs opérations identiques peuvent être faites simultanément. En supposant que l'évaluation se conforme strictement au parenthésage, montrer que l'évaluation de

(A+(B+(C+(D+(E+(F+(G+H))))))) est 2,33 fois plus lente que celle de

((A+B)+(C+D))+((E+F)+(G+H))

On suppose la division 3 fois plus lente que la multiplication et l'addition. Comment calculer en 5 unités de temps $(A+B)/(C+D)+E \cdot F+G \cdot H$?

6.7.2

On considère le tri d'une table T[1:n] ainsi défini :

- on compare T[1] à chacun de ses successeurs, en les permutant si T[1] est le plus grand, de façon à avoir *finalement* T[1] = min(T[1 : n])
- on installe ensuite de même min(T[2 : n]) en T[2], puis min(T[3 : n]) en T[3] ...
- on termine sur T[n – 1 : n].

Montrer que la durée d'un tel tri est en n^2 : ($t(tri) = \Theta n^2$).

6.7.3

On s'intéresse à une fonction de fusion de séquences croissantes définie par:

```
interclass(A : int*, B : int*) → int* =
    {       vide(A) → B ;
            vide(B) → A ;
            tete(A) ≤ tete(B) → tete(A) & interclass(reste(A), B) ;
            tete(B) & interclass(A, reste(B))
    }
```

Montrer que interclass(<0, 1, 1, 2, 3, 5, 8, 13>, <0, 1, 4, 9, 16>) = <0, 0, 1, 1, 1, 2, 3, 4, 5, 8, 9, 13, 16>.

Montrer que si card(A) = m, card(B) = n, alors t(interclass) = O(m + n).

Soient C : int* et D : int* des suites non triées d'entiers, et tri une fonction de tri en n^2. Montrer que, le plus souvent, tri(C & D) est plus lent que interclass(tri(C), tri(D)).

6.8 PROBLÈMES

6.8.0

Etudier la complexité de la fonction f_{91}: $\mathbb{Z} \to \mathbb{N}$:

$$f_{91}(n) = \{n<0 \to f_{91}(n + 17) \; ; \; n>3 \to f_{91}(n - 5) \; ; \; 91\}$$

En proposer une version optimale.

6.8.1

On considère des matrices $n \times n$; montrer que le calcul d'une matrice somme de deux autres a une durée en n^2, et que le calcul d'une matrice produit de deux autres a une durée en n^3.

6.8.2 Application multiprocesseurs (d'après Jean Ichbiah)

On considère un traitement formé

- d'une partie S, irrévocablement séquentielle,
- d'une partie P, totalement parallélisable (ex : traitement identique sur les 30 030 cases d'un même tableau).

Dans une version séquentielle du traitement, on a T = (t(P) + t(S)).

On veut passer à une version parallèle du traitement en éclatant P sur n processeurs, chacun exécutant 1/n du processus.

Quelle est la limite de la durée d'exécution quand n tend vers l'infini ?

En réalité, il faut tenir d'un compte d'une pénalité de gestion (lancement / arrêt) d'une durée ε par processus lancé en parallèle. Montrer que la durée d'exécution passe par un minimum en fonction du nombre de processeurs. Donner t_{opt} et n_{opt}. Cas où $t(P) = 0{,}7 \cdot T$ avec $\varepsilon = 0{,}001 \cdot T$. Indiquer le nombre de processeurs recommandé, car menant à des durées au plus égales à 1,05 t_{opt}.

6.8.3

Montrer que

$$C1(k, n) \equiv \{ (k = 0 \vee k = n) \rightarrow 1; C1(k-1, n-1) + C1(k, n-1)\}$$

et

$$C2(k, n) \equiv \{(k = 0 \vee k = n) \rightarrow 1; C2(k-1, n-1)\cdot n/k \}$$

sont deux définitions équivalentes, et que la seconde est strictement moins complexe que la première.

6.8.4

Une fonction est-elle représentable par une table ?

Si oui, faut-il préférer une table ou un algorithme ?

6.9 LECTURES

BENZAKEN C., Aperçu sur la complexité, § 3.5 in *Systèmes formels ; introduction à la logique et à la théorie des langages,* Masson, 1991.

FROIDEVAUX Ch., GAUDEL M.C. & SORIA M., *Types de données et algorithmes,* Ediscience International, 1994 (surtout le chapitre II).

MORRISON Ph., MORRISON Ph., *Les puissances de dix. Les ordres de grandeur dans l'Univers,* Belin, 1985.

VOROBIEV N., *Caractères de divisibilité ; suite de Fibonacci*, Collection Initiation aux Mathématiques, Editions de Moscou, 1973.

P. WEISS & X. LEROY, *Le langage CAML,* InterEditions, 1993.

DEUXIÈME PARTIE

GRAPHES

Devant un grand nombre de situations, une vieille habitude pousse l'homme à tracer sur le papier des points *représentant des individus, des localités, des corps chimiques etc., joints entre eux par des lignes ou* des flèches *symbolisant une certaine relation. Ces schémas se rencontrent partout sous des noms différents : sociogrammes (psychologie), simplexes (topologie), circuits électriques (physique), diagrammes d'organisation (économie), réseaux de communication, arbres généalogiques, etc.; D. KÖNIG, sans doute le premier, a proposé d'appeler* graphes *de tels schémas et d'étudier systématiquement leurs propriétés.*

Il est très remarquable que des disciplines aussi variées aient à utiliser des théorèmes analogues ; on sait que la notion de matrice d'incidence *introduite par KIRCHHOFF pour étudier les circuits électriques, a été reprise en topologie par Henri POINCARÉ pour fonder son* analysis situ *; la notion de* point d'articulation *connue depuis longtemps en sociologie, est apparue plus récemment en électronique ; et de tels exemples sont innombrables... La théorie des graphes, pour s'appliquer à des domaines aussi variés, devra être essentiellement abstraite et formalisée.*

En fait, les concepts fondamentaux tels que chaîne, chemin, centre, *s'ils sont définis d'une façon abstraite, restent en rapports constants avec des réalités graphiques, faciles à reconnaître lorsque le schéma est tracé. C'est pour cette raison que la théorie des graphes ne doit pas être confondue avec la théorie des relations, dont les préoccupations sont autres. [...]*

En exposant ici la théorie des graphes, notre but a été de mettre entre les mains du lecteur un outil mathématique s'appliquant tant aux sciences du comportement, à la théorie de l'information, à la cybernétique, aux jeux, aux réseaux de transport qu'à la théorie des ensembles, aux matrices ou à d'autres disciplines purement abstraites.

C. Berge,
Théorie des graphes et ses applications, Introduction (extraits),
Dunod, 1958.

CHAPITRE 7

DES POINTS ET DES FLÈCHES

7.1 PRÉSENTATION

DÉFINITION 7.1 – On appelle ***graphe (orienté)*** et on note G = <X, U> un couple formé :

- d'un ***ensemble*** X, en général fini, ***de*** N ***points*** ou ***sommets*** du graphe,
- d'un ***ensemble*** $U \subseteq X \otimes X$ de M *paires ordonnées de sommets*, ou ***arcs***.

NOTES

- U est assimilable à une relation binaire interne à X.
- Sur un dessin du graphe, X est représenté par un ensemble de points séparés, et chaque arc (x y) sera représenté par une ***flèche*** d'origine x et d'extrémité y.
- On donne le nom particulier de ***boucles*** aux arcs ayant un même point comme origine et comme extrémité.

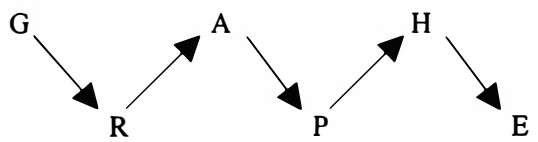


Fig. 7.1 Un graphe, ses sommets, ses arcs.

EXEMPLES

Partant de P ensemble de Personnes,

- la relation interne « enfant de » mène à un graphe sans boucle dit « arbre généalogique » ;
- la relation interne « aime » mène à un graphe pouvant posséder des boucles.

DÉFINITION 7.2 – Un ***graphe*** noté G = <X, Γ> sera défini par :

- un *ensemble* X de N *points* ou *sommets* du graphe,
- une *application*[1] $\Gamma : X \rightarrow 2^X$ associant à chaque point de X comme origine, un sous-ensemble éventuellement vide de X comme image.

NOTE. On passe de l'une à l'autre définition par : $(x\ y) \in U \equiv y \in \Gamma x$.

7.2 DEGRÉS & DEMI-DEGRÉS

DÉFINITION 7.3 – Pour un sommet i, nous appellerons

---

[1] Dite plus rarement *application multivoque* de X dans X.

- ***sortance***[2] $d^+(i)$ le nombre d'arcs ayant ce sommet pour origine (ou *partant* de ce sommet) ;
- ***entrance***[3] $d^-(i)$ le nombre d'arcs ayant ce sommet pour extrémité (ou *arrivant* en ce sommet) ;
- ***degré*** la somme $d(i) = d^+(i) + d^-(i)$ des entrance et sortance.

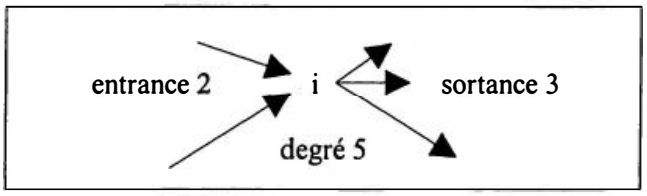


Fig. 7.2 Degré et demi-degrés d'un sommet i.

7.3 PRÉDÉCESSEURS, SUCCESSEURS, ADJACENTS

DÉFINITION 7.4 – Soit x un sommet d'un graphe $G = \langle X, \Gamma \rangle$.

Γx est l'ensemble des ***successeurs*** de x, et $d^+(x) = |\Gamma x|$.
$\Gamma^{-1} x$ est l'ensemble des ***prédécesseurs*** de x, et $d^-(x) = |\Gamma^{-1} x|$.
2 sommets qui partagent un arc sont dits ***sommets adjacents.***

PROPOSITION 7.1 – Le nombre de sommets adjacents à x est *au plus*

$$d(x) = d^+(x) + d^-(x) = |\Gamma x| + |\Gamma^{-1} x|$$

valeur atteinte quand ces ensembles sont disjoints.

7.4 SOMMETS INITIAUX, TERMINAUX, ISOLÉS

DÉFINITION 7.5 – Un sommet sera dit

- ***sommet initial*** s'il n'a pas de prédécesseur autre que lui-même,
- ***sommet terminal*** s'il n'a pas de successeur autre que lui-même,
- ***sommet isolé*** s'il n'a pas de sommet adjacent autre que lui-même.

NOTE. Pour un déplacement dans un graphe, les sommets terminaux se comportent en pièges, tandis que les sommets isolés ou initiaux sont inaccessibles.

7.5 SOUS-GRAPHE

DÉFINITION 7.6 – Soit un graphe $G = \langle X, U \rangle$. On appelle ***sous-graphe*** de G tout graphe $G' = \langle X', U' \rangle$ tel que X' soit un sous-ensemble de X, et U' le sous-ensemble correspondant de U :

$$X' \subseteq X,\ U' = U \cap (X' \otimes X')$$

---

[2] Ou *demi-degré extérieur.*

[3] Ou *demi-degré intérieur.*

EXEMPLE. Si G est le graphe des routes de France, en sera sous-graphe toute restriction à une partie de la France : sous-graphe des routes de Bretagne, Dauphiné ou Savoie.

NOTE. Un sous-graphe de G correspond à une *restriction* de la relation associée à G, au sens de la section 4.3.

7.6 GRAPHE PARTIEL

DÉFINITION 7.7 – Soit un graphe G = <X, U>. On appelle ***graphe partiel*** de G tout graphe G′ = <X, U′> tel que U′ ⊆ U.

EXEMPLE. Si G est le graphe des routes de France, on pourra considérer le *graphe partiel G′ des routes nationales.* Le graphe des routes nationales en Bretagne G'_B est alors un sous-graphe de G′, et un sous-graphe partiel de G.

7.7 GRAPHE COMPLET, CLIQUES

DÉFINITION 7.8 – Un ***graphe complet*** G = <X, X⊗X> comprend tous les arcs possibles ; il correspond à la relation universelle U. On note K_n le graphe complet à n points (abrégé pour n^{ième} ***figure de Kuratowski***).

On appelle ***clique*** un *sous-graphe complet.*

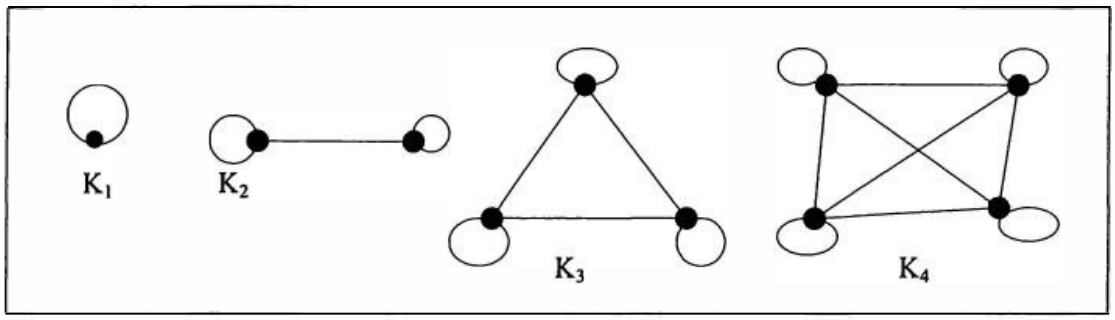


Fig. 7.3 Premières figures de Kuratowski.

7.8 REPRÉSENTATIONS

DÉFINITION 7.9 – Un graphe G = <X, U> peut être représenté par une matrice booléenne *carrée* [A], dite ***matrice d'adjacence***, formée à raison :

- d'une i-ème ligne par élément $x_i \in X$ considéré comme *origine,*
- d'une j-ième colonne par élément $x_j \in X$ considéré comme *image*
- d'éléments a_{ij} = *(si* $(x_i, x_j) \in U$ *alors* 1 *sinon* 0).

→ *Encombrement en* N^2

EXEMPLE. Soit un graphe $G_1 = <X_1, U_1>$, avec X_1 = {a b c d e} et U_1 = {(a c)(b a)(b d)(c b)(c e)(d c)(e a)(e d)} (fig. 7.4). Sa matrice d'adjacence est donnée tableau 7.1.

NOTE. La *matrice d'adjacence d'un graphe* G = <X, U> est identique à la *matrice caractéristique de la relation* U interne à X (§ 4.1.1).

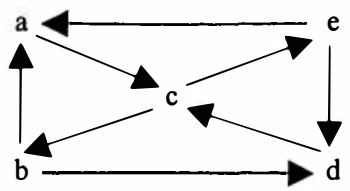


Fig. 7.4 Graphe G_1.

Tableau 7.1 Matrice d'adjacence du graphe G_1.

	a	b	c	d	e
a	o	o	1	o	o
b	1	o	o	1	o
c	o	1	o	o	1
d	o	o	1	o	o
e	1	o	o	1	o

DÉFINITION 7.10 – Un graphe G = <X, U> peut être représenté en associant à chaque sommet a_i la liste Γa_i de ses successeurs, correspondant aux 1 de la i-ème ligne de la matrice d'adjacence

- c'est une ***liste d'adjacence*** si on forme une liste de listes (sommet successeurs), comme dans L = ((a c)(b a d)(c b e)(d c)(e a d)) pour représenter le graphe G_1 ci-dessus ;
- c'est une ***table d'adjacence*** si on utilise des tables.

Tableau 7.2 Table d'adjacence du graphe G_1.

	a	b	c	d	e			
premier arc	1	2	4	6	7			
n° d'arc	1	2	3	4	5	6	7	8
arcs(fins)	c	a	d	b	e	c	a	d

Les listes auront un encombrement en M + N.

Pour les tables, l'encombrement

- est en N·sup(sortance(x)) si on utilise une table à deux dimensions, à raison d'une ligne par sommet, chaque ligne comprenant le liste des successeurs de son sommet ;
- est en M + N avec un système à 2 tables contenant :
  - la première, pour chaque sommet, le n° de son premier arc, les arcs étant supposés numérotés « sans trou »,
  - la seconde, pour chaque arc, son extrémité. (tab. 7.2).

DÉFINITION 7.11 – La ***matrice d'incidence*** d'un graphe G = <X, U> est une matrice rectangulaire formée à raison :

- d'une i-ème ligne par point $x_i \in X$ considéré,
- d'une j-ième colonne par arc $u_j \in U$ considéré,
- *d'éléments e_{ij} valant*

  1 si x_i origine de u_j
  -1 si x_i extrémité de u_j
  0 sinon.

→ *Encombrement en M· N*

Il sera équivalent, parfois préférable, de considérer une *table d'incidence,* d'encombrement 2·M, comportant

- autant de colonnes qu'il y a d'arcs,
- 1 ligne *origine* et 1 ligne *extrémité.*

NOTE. On observera une dualité classique entre encombrement et temps d'accès ; pour ce dernier, on distinguera

- la durée pour décider si un couple désigne un arc du graphe,.
- la durée pour trouver les successeurs d'un sommet...

Tableau 7.3 Matrice d'incidence de G_1.

arcs / sommets	1	2	3	4	5	6	7	8
a	+ 1	– 1	o	o	o	o	– 1	o
b	o	+ 1	+ 1	– 1	o	o	o	o
c	– 1	o	o	+ 1	+ 1	– 1	o	o
d	o	o	– 1	o	o	+ 1	o	– 1
e	o	o	o	o	– 1	o	+ 1	+ 1

Tableau 7.4 Table d'incidence de G_1.

arcs	1	2	3	4	5	6	7	8
origine	a	b	b	c	c	d	e	e
extrémité	c	a	d	b	e	c	a	d

7.9 PROPRIÉTÉS DES RELATIONS BINAIRES ET GRAPHES ASSOCIÉS

A chaque propriété d'une relation binaire interne correspondra bijectivement une propriété spécifique de son graphe, qui recevra en général le même qualificatif.

Ainsi, graphe réflexif/ transitif/ symétrique/ ... signifiera graphe d'une relation réflexive/ transitive/ symétrique/...

7.9.1 Graphe complémentaire

Le complément d'une relation $R \subseteq X \otimes X$ est $R' = (X \otimes X) - R$; de même :

DÉFINITION 7.12 – A un graphe $G = \langle X, U \rangle$ on peut associer un ***graphe complémentaire*** $G' = \langle X, U' = (X \otimes X - U) \rangle$.

7.9.2 Graphe réflexif

DÉFINITION 7.13 – Un ***graphe réflexif*** sera un graphe dont *chaque sommet porte une boucle* ; un ***graphe irréflexif***, un graphe sans boucle (i.e. dont *aucun sommet ne porte de boucle*).

PROPOSITION 7.2 – La diagonale principale d'une matrice d'adjacence

- est à 1 si le graphe est *réflexif*,
- est à 0 si le graphe est *irréflexif*,
- est mixte si le graphe est *non réflexif*.

7.9.3 Graphe transitif et son squelette

DÉFINITION 7.14 – Un ***graphe transitif*** sera tel qu'à toute paire d'arcs consécutifs (a, b) et (b, c) correspondra un troisième arc (a, c).

NOTE. Un graphe transitif sera souvent dit *touffu*, ses arcs souvent très nombreux le rendant difficile à dessiner ; au graphe d'une relation R transitive, réflexive ou irréflexive, on préfèrera donc souvent son « squelette », plus lisible.

Soit R une relation transitive, réflexive ou irréflexive, et sa *relation-squelette* ρ, obtenue (§ 4.2.3).

- pour R irréflexive en enlevant tout arc (x z) s'il existe un arc (x y) et un arc (y z) : alors $\rho = R - R^2$;
- pour R réflexive, à partir de $R' = R - I$; alors $\rho = R' - R'^2$. De même :

DÉFINITION 7.15 – On appelle ***squelette*** d'un graphe transitif (ir)réflexif G = <X, R> le graphe Γ = <X, *ρ*> correspondant à la relation-squelette *ρ*(R).

EXEMPLE. Au graphe transitif et réflexif G_0 de la figure 4.1 correspond le squelette Γ_0 donné figure 4.2.

NOTE. La connaissance du caractère (ir)réflexif du graphe d'origine G est nécessaire pour restituer celui-ci à partir du squelette Γ.

7.9.4 Graphe symétrique

DÉFINITION 7.16 – Un ***graphe symétrique*** est formé de sommets liés par des ***arêtes***, *version non orientée des arcs, une arête (x y) étant assimilable à un couple d'arcs (x y), (y x).*

PROPOSITION 7.3 – Si [R] est la matrice carrée d'adjacence d'un graphe symétrique, elle est symétrique par rapport à la diagonale principale, et $[R] = [R]^T$, où $[R]^T$ désigne la transposée de [R].

7.9.5 Graphe inverse

DÉFINITION 7.17 – L'***inverse*** G^{-1} d'un graphe G s'obtient en renversant le sens des arcs.

$$G = \langle X, U\rangle \rightarrow G^{-1} = \langle X, V = \{(a, b) \mid (b, a) \in U\}\rangle$$

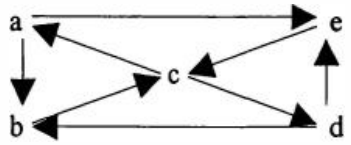


Fig. 7.5 G_1^{-1}, graphe inverse de G_1.

PROPOSITIONS

7.4 – L'inversion d'un graphe est involutive (cf. prop. 2.6).

$$(G^{-1})^{-1} \equiv G$$

7.5 – Soit un graphe G représenté par une matrice [R] et son inverse G^{-1} représenté de même par $[R^{-1}]$

- Si [R] est une matrice carrée d'adjacence : $[R^{-1}] = [R]^T$.
- Si [R] est une matrice d'incidence : $[R^{-1}] = -[R]$.

7.9.6 Symétrisé d'un graphe

Lorsqu'on traite le graphe d'une relation R non symétrique, l'orientation n'est pas toujours pertinente.

DÉFINITION 7.18 – On appellera ***symétrisé d'un graphe*** orienté G le graphe symétrique G_S obtenu en faisant abstraction des orientations dans G, i.e. en associant à chaque *arc* de G une *arête* de G_S.

G_S ainsi obtenu est le graphe (symétrique) associé à la symétrisée de R : $R_S = R \cup R^{-1}$.

PROPOSITION 7.6 – La matrice d'adjacence $[R_s]$ d'un graphe G_S symétrisé d'un graphe G, s'obtient à partir de la matrice d'incidence [R] de G par :

$$[R_s] = [R] + [R^{-1}] = [R] + [R]^T, \qquad \text{où + désigne le } \textit{ou booléen}.$$

7.10 PROPRIÉTÉ FAIBLE / PROPRIÉTÉ FORTE

DÉFINITION 7.19 – Un graphe donné sera dit ***fortement*** P, s'il s'agit d'un graphe orienté pour lequel la propriété P a été établie *en respectant les orientations des arcs.*

Un graphe donné sera dit ***(faiblement)*** P, s'il s'agit d'un graphe orienté pour lequel la propriété P a été établie *en négligeant les orientations des arcs*, par exemple en raisonnant sur son symétrisé.

Les deux notions se confondent pour un graphe symétrique (cf. aussi chap. 10).

7.11 GRAPHE VALUÉ

DÉFINITION 7.20 – On appelle ***graphe valué*** une structure $G = \langle X, U, V, \mu : U \rightarrow V \rangle$ formée

- d'un graphe de base $g = \langle X, U \rangle$
- d'un ensemble V dit ***ensemble de valuation***,
- d'une *application* μ associant à chaque arc de U une valeur de V.

Les représentations associées sont souvent similaires à celles des graphes simples, de même que les traitements.

EXEMPLE. Au lieu qu'une matrice booléenne dise s'il y a ou non une liaison routière directe entre x et y, on peut utiliser

- une matrice des *distances*,
  - en remplaçant chaque « 1 » concernant un arc existant par la distance correspondante ;
  - en codant ∞ toute absence d'arc pour marquer l'impossibilité en termes de distance.
- une matrice des *débits*,
  - en remplaçant chaque « 1 » concernant un arc existant par le débit admissible (en véhicules/heure) sur le tronçon concerné ;

- en codant 0 un arc absent ou temporairement fermé pour marquer l'impossibilité en termes de débit.

- une matrice d'énonciation,
  - en remplaçant chaque « 1 » concernant un arc existant par le nom de la route sur le tronçon concerné (N7, A89, D317e ...) ;
  - en codant 0 par un symbole spécial marquant l'inexistence (ω, $\perp$ ...).

Cette partie Graphes passera souvent des graphes simples à des graphes ayant une valuation simple à traiter, lorsqu'on pourra le faire à peu de frais. C'est d'ailleurs ainsi que sont nés, de façon plus ou moins naïve, de nombreux algorithmes sur les graphes.

Dans la partie Algèbre, une approche plus générique sera présentée au chapitre Dioïdes (chap. 14) avec application non numérique aux Algèbres de Kleene (chap. 16).

7.12 EXERCICES

7.12.1

Considérer l'ensemble X = {1, 2, 3, 4, 5, 6, 7, 8}, et lui associer un graphe G_{div} = <X, U>, tel que (x y) $\in$ U si et seulement si x est un diviseur de y.

Vérifier que tout sommet de G_{div} est porteur d'une boucle. Pourquoi ?

Vérifier que G_{div} est transitif.

7.12.2

Considérer l'ensemble D des droites d'un plan P, et lui associer un graphe G_{perp} = <D,U>, tel que (x y) $\in$ U si et seulement si x est perpendiculaire à y.

Vérifier qu'aucun sommet de G_{perp} n'est porteur de boucle. Pourquoi ?

Vérifier que G_{perp} est intransitif.

7.12.3

Considérer un ensemble de personnes P, et lui associer un graphe G = (P,U), tel que (x y) $\in$ U si et seulement si x est en excellents ou bons termes avec y. Le graphe est-il réflexif ? symétrique ? transitif ?

7.13 PROBLÈMES

7.13.0 Les Antiquités de Madame Marcia

« Madame Marcia habite, avec son mari et son fils, un appartement de trois pièces au rez-de-chaussée droite. Son magasin est au rez-de-chaussée également, mais à gauche, entre la loge de la concierge et l'entrée de service. Madame Marcia n'a jamais établi de distinction réelle entre les meubles qu'elle vend et ceux dans lesquels elle vit, ce qui fait qu'une part importante de ses activités consiste à transporter meubles, lustres, lampes, pièces de vaisselle et objets divers entre son appartement, son magasin, son arrière-boutique et sa cave. Ces échanges, qui sont suscités aussi bien par des occasions propices de vente que d'achat (il s'agit alors de faire de la place) que par des inspirations subites, des lubies, des caprices ou des dégoûts, ne se font par hasard et n'épuisent pas les douze possibilités de permutation qui pourraient se faire entre ces quatre lieux [...] ; ils obéissent strictement au schéma [...]: quand Madame Marcia achète quelque chose, elle le met chez elle, dans son appartement ou sa cave ; de là, le dit objet peut passer dans l'arrière-boutique, et

de l'arrière-boutique dans le magasin ; du magasin enfin il peut revenir – ou parvenir, s'il venait de la cave – dans l'appartement. Ce qui est exclu, c'est qu'un objet revienne dans la cave, ou repasse du magasin dans l'arrière-boutique, ou de l'arrière-boutique dans l'appartement, ou enfin passe directement de la cave à l'appartement »

(D'après Georges Perec, *La vie mode d'emploi*, Hachette 1978, repris de Jean Julo, *Représentation des problèmes et réussite en mathématiques*, coll. Psychologies, Presses Universitaires de Rennes, 1994).

Définir la liste des 4 lieux de stockage en cause, et vérifier le nombre de permutations indiquées.

Partant du passage « schéma ... appartement », définir le graphe des transitions directes d'un lieu à un autre.

Le vérifier ensuite avec la suite « Ce qui est exclu... ».

7.13.1

On parle souvent de graphes *clairsemés* ou *touffus*. M désignant le nombre d'arcs et N le nombre de sommets, bâtir un indicateur en (M, N) distinguant ces cas.

7.14 THÈMES DE RÉFLEXION

7.14.0

Utilisation d'un graphe pour visualiser les liens généalogiques dans une population.

7.14.1

Utilisation d'un graphe pour visualiser les liens logiques dans une théorie.

7.14.2

Utilisation d'un graphe pour visualiser les actions possibles dans un jeu.

7.14.3

Utilisation d'un graphe pour visualiser l'ordonnancement des tâches d'un projet.

7.14.4

Utilisation d'un graphe pour visualiser l'analyse syntaxique d'une phrase.

7.14.5

Utilisation d'un graphe pour visualiser la structure d'un programme ; si on assimile les procédures aux sommets, quelle signification donner aux arcs ?

7.15 LECTURES

AHO A., ULLMAN J., *Concepts fondamentaux de l'informatique*, chap. IX, Dunod, 1993.

BERGE C., *Théorie des graphes et ses applications*, Coll. Universitaire de Mathématiques n° 2, Dunod, 1958/63.

BERGE C., *Graphes,* Gauthier-Villars / Bordas, 1970/73/83.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien, tome I, chap. II,* Coll. Programmation, Ed. Gauthier-Villars, 1973.

FEJER P.A., SIMOVICI D.A., *Mathematical Foundations of Computer Science,* vol. I, chap. II & III, Coll. Texts and Monographs on Computer Science, Springer Verlag.

GONDRAN M., MINOUX M., *Graphes et algorithmes,* chap. I, Coll. Direction des Etudes & Recherches EdF, Eyrolles, 1979/85/95.

KAUFMANN A., *Des points et des flèches,* Dunod-poche, 1969.

PICARD C.F., *Graphes & Questionnaires,* tome I : Graphes, chap. I et II, Coll. Programmation, Ed. Gauthier-Villars, 1972.

REINHARDT F. & SOEDER H., *Atlas des mathématiques,* pp. 250-255, La Pochothèque, 1997.

collectif, *Petite encyclopédie des mathématiques*, chap. XXXVI, Edition Pagoulatos, Didier distr.

NOTE.

Le concept de graphe développé dans cette partie est issu de KÖNIG, *Theorie der endlichen und unendlichen Graphen,* Leipzig, 1936, introduit en France par [BERGE 58].

La littérature qui s'est ensuite développée en français est dominée par :

C. BERGE, à qui on doit des études sur les jeux et les réseaux de transport, puis une théorie recentrée sur les hypergraphes (1973).

B. ROY, auteur du premier grand ouvrage pratique ; inventeur de la Méthode des potentiels (ordonnancement) ; plus récemment, actif dans le domaine de la décision multicritères.

C.F. PICARD, théoricien des questionnaires (préfigurant les menus).

J. KUNTZMANN, qui s'intéressa aux nombreux emplois des graphes bipartis en automatique, établit leur équivalence avec les hypergraphes, et introduisit en 1972 les dioïdes pour un traitement généralisé des problèmes de chemin, y compris la question des fermetures transitives en logique multivaluée.

DERNIAME et PAIR, qui mirent en relief les possibilités de généricité.

GONDRAN et MINOUX, dont la somme est régulièrement rééditée depuis 1976...

CHAPITRE 8

CHEMINS & CIRCUITS

Si un arc rend compte d'une liaison ou influence directe, un chemin rendra compte d'une liaison ou influence indirecte ou généralisée. Un circuit, chemin bouclé sur lui-même, rendra compte des phénomènes auto-entretenus, et pourra constituer un piège dans la recherche de chemins.

Plus généralement, la *métaphore du labyrinthe* met ces concepts au service de la *résolution de problèmes*. Pour cela, on assimile les situations possibles dans un univers à des points ; pour chaque situation, à chaque action possible correspondra un arc marqué du nom de l'action et représentant la transition qu'elle induit. Un problème étant alors défini par une situation de départ et une (des) situation(s) finale(s) exprimant un but, une *solution* est un chemin amenant de ce départ à ce but, énonçant une suite admissible d'actions réalisant pas à pas la transition désirée dans le labyrinthe des possibles. La multiplicité de telles solutions permettant d'éliminer les moins agréables et/ou les moins réalistes, à chaque chemin retenu devrait correspondre un *plan d'action* final adéquat, libre de tout bouclage inutile.

8.1 LES CHEMINS

DÉFINITION 8.1 – Soit un graphe G = <X, U> ; on appelle ***chemin*** de G toute suite de sommets $x_0, x_1, x_2, \ldots, x_n$, telle que chaque couple $(x_i\, x_{i+1})$ soit un arc de U.

Un *chemin* formé de n arcs successifs sera dit ***chemin de taille n***.

EXEMPLES

Tout arc est un chemin de taille 1.

G_1 (fig. 7.4) comporte des chemins de taille 1 (a c), 2 (a c b), 3 (a c b d), 4 (e a c b d).

Un chemin de taille 9 est donné dans la figure.

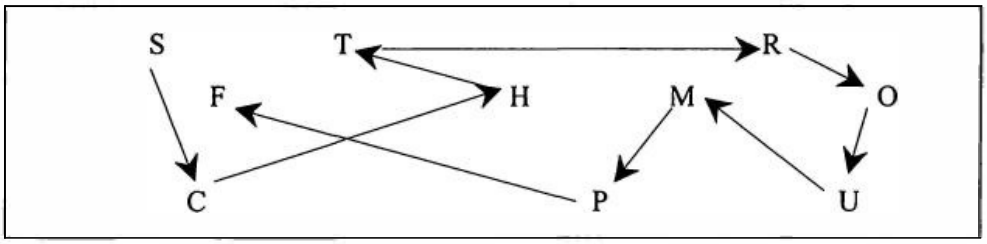


Fig. 8.1 SCHTROUMPF, chemin de taille 9.

PROPOSITION 8.1 – Dans tout graphe transitif, s'il existe un *chemin* de x à y, alors il existe un *arc* de x à y, dit ***corde*** du chemin.

PREUVE. Supposons cette propriété vraie pour les chemins de taille au plus égale à n.

Tout chemin de taille n+1 est un chemin $(x_0, x_1, x_2, \ldots x_{n+1})$ formé d'un chemin $s = (x_0, x_1, x_2, \ldots x_n)$ de taille n suivi d'un arc $(x_n\, x_{n+1})$.

Au chemin s nous pouvons associer un arc $(x_0\ x_n)$ par hypothèse ; l'arc $(x_n\ x_{n+1})$ et la transitivité du graphe imposent alors l'existence d'un arc $(x_0\ x_{n+1})$.

La propriété est donc vraie pour les chemins de taille n+1 lorsqu'elle est vraie pour les chemins de taille n.

Or la propriété est vraie pour les chemins de taille 1, formés d'un seul arc.

Elle est donc toujours vraie. ■

NOTE. Cette propriété manifeste la *puissance* du concept de transitivité.

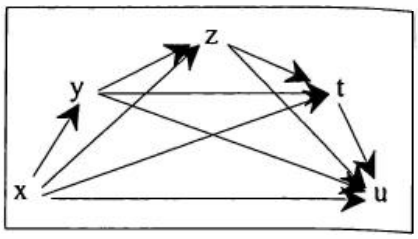


Fig. 8.2 Graphe transitif.

PROPOSITION 8.2 – Soit R la relation sur X représentée par le graphe G = <X, R> ; sont équivalents les énoncés :

il existe dans G un chemin de *taille k* de x à y
≡ (x y) satisfait la relation R^k
≡ y ***successeur au rang k*** de x : $y \in \Gamma^k x$
≡ x ***prédécesseur au rang k*** de y : $x \in \Gamma^{-k} y$

On verra que la composition de relations est un produit associatif [1] : l'ensemble $\mathcal{R}_X$ des relations sur X doté de la composition supporte alors une notion justifiée de *puissance de composition* j, en posant $R^1 = R$ et $R^0 = I_X$ pour toute relation R sur X.

On peut donc écrire $R^{m+n} = R^m \bullet R^n$, avec la signification :

$$x\, R^{m+n}\, y \equiv (\exists z\ x\, R^m z \wedge z\, R^n y)$$

En interprétant $x\, R^k\, y$ comme l'existence d'un chemin de taille k de x à y, cette équivalence signifie tout naturellement qu'il existe un chemin de taille m + n de x à y *ssi* il existe au moins un point z tel qu'il existe un chemin de taille m de x à ce z, et un chemin de taille n de ce z à y.

PROPOSITION 8.3 – Soit R la relation sur X représentée par le graphe G = <X, R>. Les énoncés suivants sont équivalents :

il existe dans G un chemin de x à y
≡ (x y) satisfait la relation R*
≡ y ***descendant de*** x : $y \in \Gamma^* x$
≡ x ***ancêtre de*** y : $x \in \Gamma^{-1} * y$

R*, Γ* se présentent comme la *généralisation,* tous rangs confondus, des R et Γ (cf. définitions 9.1 à 9.3).

EXEMPLE. Pour des personnes,

si	x R y	signifie	x père_ou_mère_de y, alors
	$x\, R^2 y$	signifie	x grand_père_ou_grand_mère_de y
	$x\, R^3 y$	signifie	x arrière_grand_père_ou_arrière_grand_mère_de y
	x R*y	signifie	x ancêtre_de y.

[1] Si besoin, voir les § 12.5.2 (associativité) et 13.1.2 (puissances de composition).

8.2 CIRCUITS ET BOUCLES DANS UN GRAPHE

DÉFINITION 8.2 – Soit un graphe G = <X, U> ; on appelle ***circuit*** de G tout chemin $x_0, x_1, x_2, \ldots x_{n-1}, x_0$ dont le premier sommet est aussi sommet final.

Un tel circuit a pour ***taille*** le nombre n d'arcs successifs.

EXEMPLE. G1 (fig. 7.3) comporte des circuits de taille 3 : (a c b a), (c b d c), (c e d c)...

COROLLAIRE 8.4 – Une ***boucle*** est un *circuit de taille 1.*

PROPOSITION 8.5 – Le *squelette* d'un graphe transitif est un graphe *sans corde ni boucle.*

PROPOSITION 8.6 – Entre sommets d'un graphe :

- la relation *est origine d'un chemin menant à* est *transitive,*
- la relation *est identique à ou origine d'un chemin menant à* est un *préordre,*
- la relation *est identique à ou sur un même circuit que* est une *équivalence.*

PREUVE. Soit, entre sommets d'un graphe G = <X, U>

$\sim$ la relation *est identique à, ou origine d'un chemin menant à,*

$\approx$ la relation *est identique à, ou sur un même circuit que.*

Alors,

$x \approx y \leftrightarrow (x \sim y \wedge y \sim x)$, et :

$x \approx x$ par définition,			*réflexivité de* $\approx$
$x \approx y \Rightarrow y \approx x$		par commutativité de $\wedge$	*symétrie de* $\approx$
$x \approx y,\ y \approx z$	$\Rightarrow$	$x \sim y,\ y \sim x,\ y \sim z,\ z \sim y$	ventilation
	$\Rightarrow$	$x \sim y,\ y \sim z,\ z \sim y,\ y \sim x$	recopie permutée
	$\Rightarrow$	$x \sim z,\ z \sim x$	transitivité de $\sim$
	$\Rightarrow$	$x \approx z$	*transitivité de* $\approx$

et $\approx$ est une équivalence. ■

NOTE. On peut aussi invoquer la proposition 4.31 :
L'intersection $\approx$ du préordre $\sim$ et de son symétrique est une équivalence.

Graphe-quotient

DÉFINITION 8.3 – A tout graphe G = <X, U>, on peut associer un *graphe-quotient* $G/\approx = <X/\approx, U/\approx>$, où $\approx$ désigne l'équivalence *être identique ou sur un même circuit dans G.*

$G/\approx$ est souvent plus lisible que G, dont il montre la *structure globale* (fig. 8.3).

Cependant, le graphe-quotient rassemble tous les points en une seule classe si, comme pour G_1, tout sommet est accessible de tout autre (cas des graphes fortement connexes, sect. 10.2).

8.3 CHEMINS ÉLÉMENTAIRES DANS UN GRAPHE

DÉFINITION 8.4 – Soit un graphe G = <X, U> ; on appelle ***chemin élémentaire*** de G tout *chemin de G sans circuit.* Un tel chemin est défini par une *suite de sommets* de G $x_0, x_1, x_2, \ldots, x_n$

- dont chaque couple $x_i x_{i+1}$ est un arc de U.
- dont *aucun sommet n'est répété.*

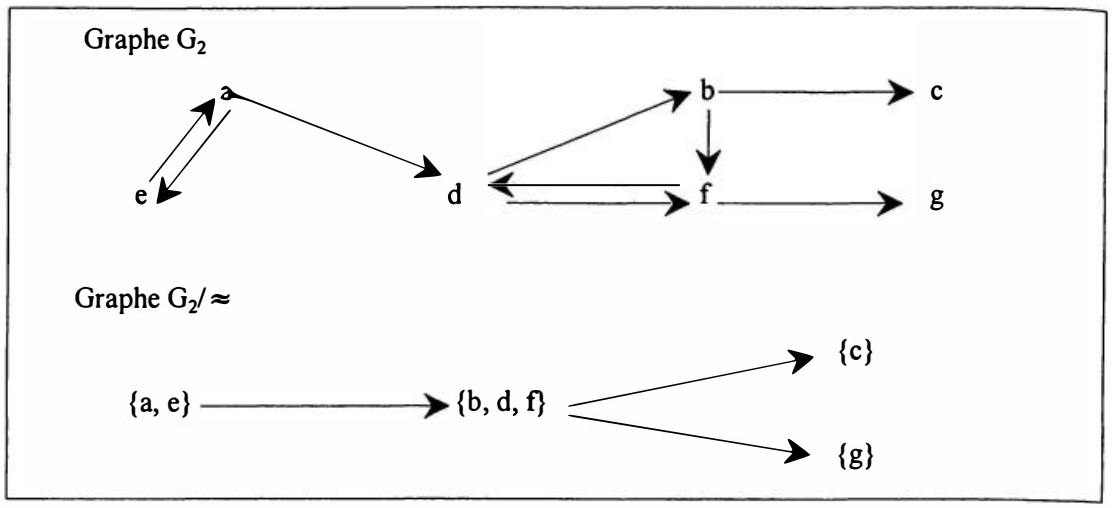


Fig. 8.3 Graphe G_2 et son graphe-quotient.

Tableau 8.1 Matrices de recherche de chemins du graphe G_2.

R	a	b	c	d	e	f	g
a	0	0	0	1	1	0	0
b	0	0	1	0	0	1	0
c	0	0	0	0	0	0	0
d	0	1	0	0	0	1	0
e	1	0	0	0	0	0	0
f	0	0	0	1	0	0	1
g	0	0	0	0	0	0	0

$\Rightarrow$

$R^{\oplus}$	a	b	c	d	e	f	g
a	1	1	1	1	1	1	1
b	0	1	1	1	0	1	1
c	0	0	0	0	0	0	0
d	0	1	1	1	0	1	1
e	1	1	1	1	1	1	1
f	0	1	1	1	0	1	1
g	0	0	0	0	0	0	0

R*	a	b	c	d	e	f	g
a	1	1	1	1	1	1	1
b	0	1	1	1	0	1	1
c	0	0	1	0	0	0	0
d	0	1	1	1	0	1	1
e	1	1	1	1	1	1	1
f	0	1	1	1	0	1	1
g	0	0	0	0	0	0	1

$R^*=R^{\oplus}\cup I$

$\Rightarrow$

R*	a	e	b	d	f	c	g
a	1	1	1	1	1	1	1
e	1	1	1	1	1	1	1
b	0	0	1	1	1	1	1
d	0	0	1	1	1	1	1
f	0	0	1	1	1	1	1
c	0	0	0	0	0	1	0
g	0	0	0	0	0	0	1

blocs diagonaux d'équivalence formés par permutation des lignes et colonnes

PROPOSITIONS. Dans un graphe à n sommets :

8.7 – Tout *chemin élémentaire* a pour taille n – 1 au plus.
8.8 – Tout ***circuit élémentaire*** (i.e. sans sous-circuit) a pour taille n au plus.
8.9 – A tout chemin c_1 on peut toujours associer (au moins) un chemin élémentaire c_2 formé exclusivement d'arcs de c_1.

PREUVE. Ce dernier point se démontre par descente récursive et couper/coller :

Tout arc (ou chemin de taille 1) qui n'est pas une boucle est un chemin élémentaire.

Soit un chemin $C = (x_0, x_1, x_2, \dots x_n)$ de taille n :

- soit C décrit un chemin élémentaire, et la propriété est établie ;

- soit C contient au moins un circuit, et il existe au moins un sommet $a \in X$ mentionné deux fois dans C ; $C = (x_0, x_1, x_2, \ldots, \mathbf{x_k = a}, x_{k+1} \ldots, x_{m-1}, \mathbf{x_m = a}, \ldots x_n)$ explicite cette répétition ; remplaçons alors dans C le circuit $(a, x_{k+1} \ldots x_{m-1}, a)$, par une seule occurrence de a : on obtient un nouveau chemin

$$C' = (x_0, x_1, x_2, \ldots x_k = a = x_m, \ldots x_n)$$

strictement plus court que C et formé exclusivement d'arcs de C.

C′ est lui-même un chemin élémentaire ; sinon, on peut lui associer un chemin C″ strictement plus court et contenant (au moins) un circuit de moins.

Le procédé se termine nécessairement, mais n'est pas déterministe, car il peut exister plusieurs sous-chemins C′ convenables pour un chemin C donné, et donc plusieurs chemins élémentaires terminaux. ■

NOTE. Qu'on puisse associer un chemin élémentaire à chaque chemin est ainsi établi ; mais ce chemin élémentaire n'est pas nécessairement unique.

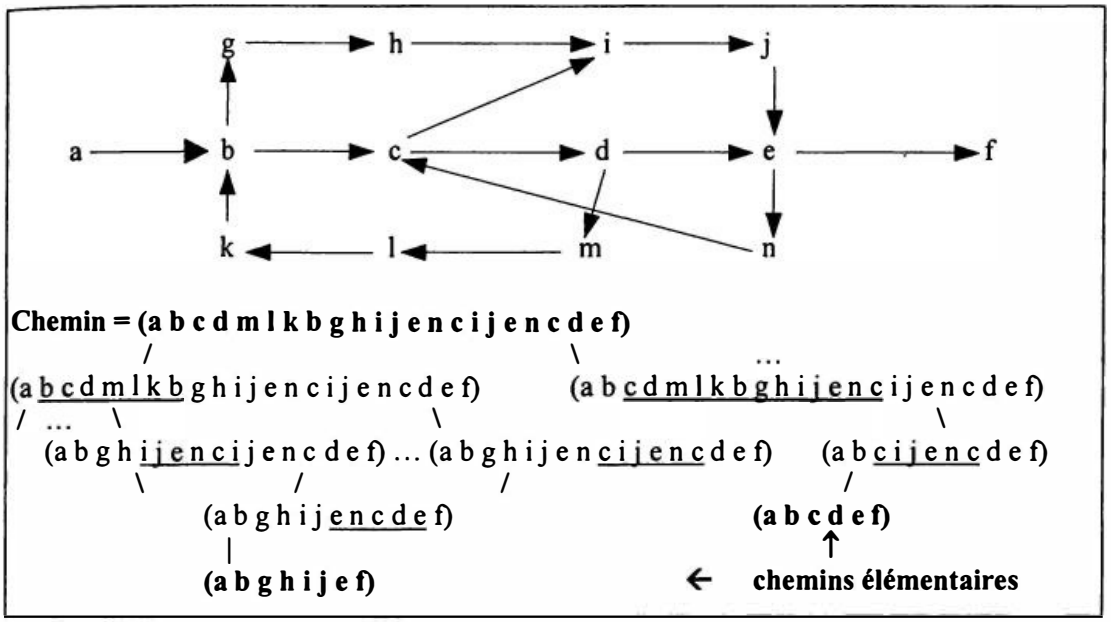


Fig. 8.4 Chemins élémentaires associés à un chemin.

8.4 RECHERCHE DE CHEMINS DANS UN GRAPHE

8.4.1 Problèmes-types

On considère qu'il existe 3 problèmes-types de recherche de chemins dans les graphes, auxquels les autres se rattachent :

P1 : *recherche du ou des chemins pour aller de x à y ;*

P2 : *recherche des chemins allant d'un x donné à tout autre sommet y ;*

P3 : *recherche des chemins allant de tout sommet x à tout sommet y .*

8.4.2 Méthodes globales

L'étude des relations associe des calculs matriciels au produit de composition de relations. En ce sens, comme l'existence d'un chemin de taille k nous renvoie à une

relation qui est la $k^{ième}$ puissance de composition de la relation du graphe, on *doit* disposer de méthodes matricielles pour régler les problèmes de chemins dans un graphe : on parlera alors de méthodes globales, *une matrice représentant un graphe en bloc.*

On associe alors à diverses classes de problèmes des (sommes de) puissance(s) de la matrice du graphe ; ce calcul permet alors de répondre à toute question de la classe.

Pour un graphe à n sommets,
les matrices auront un encombrement en n^2
leurs produits auront une durée de l'ordre de n^3,
les calculs matriciels relatifs aux chemins auront une durée en n^4 au plus.

C'est dire que les méthodes matricielles ou globales seront limitées aux graphes d'au plus quelques centaines de sommets.

Particulièrement favorables aux problèmes du 3e type, les méthodes globales seront utilisées également pour les autres problèmes lorsqu'on veut les résoudre fréquemment pour un même graphe, en introduisant

- un traitement préparatoire, matriciel, aussi lourd que nécessaire, à ne refaire que lorsque le graphe est modifié, menant à une « matrice de fermeture »,
- une « résolution » fréquente se réduisant à la consultation de la matrice de fermeture obtenue (sect. 8.5).

8.4.3 Méthodes locales

Un chemin de taille n + 1 étant un chemin de taille n auquel on ajoute un arc, des méthodes *incrémentales, pas à pas,* ou *locales* existent également.

Les *méthodes locales* (recherche par niveau ou en profondeur) seront mieux adaptées à l'*intelligence artificielle*, où l'on peut rencontrer, notamment dans certains jeux, des graphes à 10^{200} sommets et plus.

N'étant pas matériellement représentables par des structures de données, de tels graphes sont définis localement par des processus génératifs, et on parle alors de *graphes virtuels.*

Les algorithmes locaux ne s'intéresseront à tout moment qu'à une *frange* du graphe. Il y aura évidemment un prix à payer pour cette myopie nécessaire.

Les méthodes locales conviendront bien aux 2 premiers types de problèmes, et à l'hybride 2bis : *trouver les chemins allant d'un ensemble A de sommets à un ensemble B de sommets.* Elles conviendront surtout à des traitements non répétitifs, au coup par coup (par exemple, si un même graphe n'est utilisé que quelques fois).

Elles regroupent essentiellement les méthodes

- par niveaux ou en largeur *(breadth-first),* sect. 8.6,
- du meilleur d'abord *(best-first),* sect. 8.7,
- en profondeur *(depth-first),* sect. 8.8.

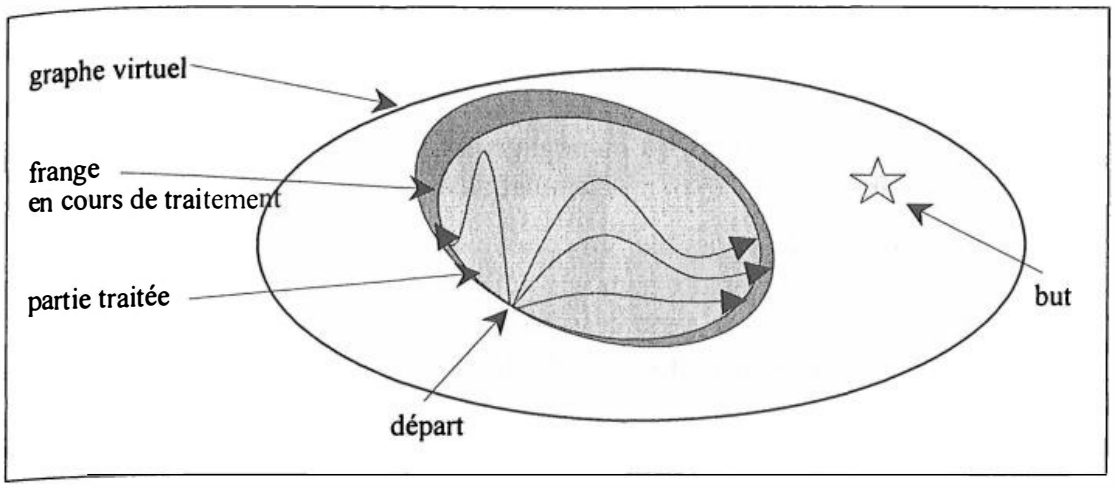


Fig. 8.5 Méthodes locales de recherche de chemin dans un graphe.

8.5 MÉTHODES MATRICIELLES

8.5.1 Chemins de taille k donnée

Soit R la relation sur X représentée par le graphe G = <X, R>. Nous avons vu (prop. 8.2) l'équivalence

il existe dans G un chemin de taille k de x à y ⇔ (x y) satisfait la relation R^k

L'existence d'un chemin de taille k entre deux points peut donc être décidée à partir de la matrice

$$[R^k] = [R]^k$$

obtenue en élevant la matrice booléenne [R] représentative du graphe, à la puissance k, dans le style relatif au produit de composition de relations, i.e. *avec OU comme somme et ET comme produit.*

Si |X| = n, chaque produit de matrice a un coût en n^3. Le coût du calcul d'une puissance k-ième dans le style de l'algorithme de Legendre est alors en $n^3 \log_2 k$, en remarquant que :

$$[R^{2p}] = [R^p]^2 \textit{ et } [R^{2p+1}] = [R^{2p}]\cdot[R].$$

Ainsi, l'évaluation de R^{37} se fait en 7 multiplications dont 5 carrages (ou élévations au carré)

$$R \Rightarrow R^2 \Rightarrow R^4 \Rightarrow R^8 \rightarrow R^9 \Rightarrow R^{18} \Rightarrow R^{36} \rightarrow R^{37}$$

8.5.2 Chemins de taille bornée

DÉFINITION 8.5 – Nous noterons

$x\ {}_kC_m\ y \equiv$ « *il existe de x à y un chemin de taille comprise entre k et m inclus* »

COROLLAIRE 8.10 – Par définition :

$$_kC_m = \bigcup_{i=k}^{m} R^i \text{ soit en termes de matrices } [_kC_m] = \sum_{i=k}^{m} [R]^k = [R]^k \cdot \sum_{i=0}^{m-k} [R]^i$$

Une somme a un coût en n^2, qu'on néglige devant le coût en n^3 des produits. Ces évaluations ont donc un coût en $m \cdot n^3$, éventuellement en $(m - k + 2 \log_2 k)n^3$.

Des améliorations sont possibles en combinant

- la méthode de Legendre pour une puissance isolée[2],
- le schéma de Horner[3] pour la somme de puissances successives,
- la relation $[\,_{k+m}C_{n+p}\,] = [\,_kC_n\,]\cdot[\,_mC_p\,]$, et sa corollaire $[\,_{2k}C_{2m}\,] = [\,_kC_m\,]^2$.

Tableau 8.2 Puissances de matrices pour la recherche de chemin.

R	a	b	c	d	e	f	g
a	0	0	0	1	1	0	0
b	0	0	1	0	0	1	0
c	0	0	0	0	0	0	0
d	0	1	0	0	0	1	0
e	1	0	0	0	0	0	0
f	0	0	0	1	0	0	1
g	0	0	0	0	0	0	0

⇒

R^2	a	b	c	d	e	f	g
a	1	1	0	0	0	1	0
b	0	0	0	1	0	0	1
c	0	0	0	0	0	0	0
d	0	0	1	1	0	1	1
e	0	0	0	1	1	0	0
f	0	1	0	0	0	1	0
g	0	0	0	0	0	0	0

R^3	a	b	c	d	e	f	g
a	0	0	1	1	1	1	1
b	0	1	0	0	0	1	0
c	0	0	0	0	0	0	0
d	0	1	0	1	0	1	0
e	1	1	0	0	0	1	0
f	0	0	1	1	0	1	1
g	0	0	0	0	0	0	0

⇒

R^4	a	b	c	d	e	f	g
a	1	1	0	1	0	1	1
b	0	0	1	1	0	1	1
c	0	0	0	0	0	0	0
d	0	1	1	1	0	1	1
e	0	0	1	1	1	1	1
f	0	1	0	1	0	1	1
g	0	0	0	0	0	0	0

Incidence de la réflexivité

PROPOSITION 8.11 – La réflexivité de la relation R entraîne que : $[_0C_k] = [R]^k$.

PREUVE. Si R est réflexif, $[_1C_1] = [R] = [I \cup R] = [I] + [R] = [_0C_1]$, puis

$[_2C_2] = [R]^2 = ([I] + [R])\cdot([I] + [R])$, du fait de la réflexivité de R

$= [I]\cdot[I] + [R]\cdot[I] + [I]\cdot[R] + [R]^2$, soit en développant

$= [I] + [R] + [R] + [R]^2$ car [I] est élément neutre de ·,

---

[2] Combinant $x^1 = x$, $x^{2p} = (x^p)^2$ et $x^{2p+1} = x^{2p}\cdot x$.

[3] Evaluant toute forme polynômiale $P_n(x)$ sous la forme $Q_{n-1}(x)\cdot x + a$, avec par exemple $x^5 + x^3 + x + 1 = (x^4 + x^2 + 1)\cdot x + 1 = ((x^3 + x)\cdot x + 1)\cdot x + 1 = ((x^2 + 1)\cdot x^2 + 1)\cdot x + 1$.

$= [I] + [R] + [R]^2 = [_0C_2]$ par idempotence de +

et, par récurrence, $[_0C_k] = [R]^k$. ■

Le procédé correspondant sera d'autant plus fort que, dans de nombreuses modélisations, le choix réflexivité / irréflexivité est arbitraire.

REMARQUES

- Pour une relation *quelconque*, l'évaluation de R^{37} par R, R^2, R^4, R^8, R^9, R^{18}, et enfin R^{36}, en 7 multiplications dont 5 élévations au carré, décide de l'existence des chemins de taille 37 *exactement* du graphe correspondant.
- Mais pour une relation *réflexive*, l'évaluation de R^{37} par le même procédé fournit finalement tous les chemins de taille 37 *au plus* du graphe correspondant. En effet, pour de telles relations $R^{37} = {}_0C_{37}$ car à tout chemin S de taille inférieure à 37 on peut associer un chemin S′ de taille 37 en rajoutant à S autant de boucles que nécessaires.

Exemple

Traitement

Reprenons le graphe G_2 défini à la figure 8.3 et au tableau 8.1, et cherchons ses chemins de taille 1 à 4.

Comment calculer sans ruse[4] *le terme* R^2 [d, f] ?

R^2 [d, f] ≡ si (∃ x R(d,x) ∧ R(x,f)) alors 1 sinon 0 ≡ $\underset{x \in X}{\text{OU}}$ (R[d,x] ET R[x,f])

Tableau 8.3 Calcul de $R^2[d, f] = \bigvee_x (R[d, x] \wedge R[x, f])$.

x	a	b	c	d	e	f	g	
R(d,x)	o	*1*	o	o	o	1	o	
R(x, f)	o	*1*	o	1	o	o	o	ou
et	o	*1*	o	o	o	o	o	⇒*1*

$R[d, b] = 1 \wedge R[b, f] = 1 \Rightarrow R^2[d, f] = 1$

On part de la ligne R[d, *] et de la colonne R[*, f] de la table R.
On forme toutes les conjonctions R[d, x]∧R[x, f].
Le résultat est un OU sur toutes ces conjonctions, i.e.

- 1 s'il y en a une et plus à 1,
- 0 s'il y en a 0.

De même, pour calculer sans ruse le terme R^{m+n} [d, f] :

- on part de la ligne R^m[d, *] et de la colonne R^n[*, f] des tables R^m et R^n respectivement ;
- on forme toutes les conjonctions R^m[d, x]∧R^n[x, f] ;
- le résultat est un OU sur toutes ces conjonctions, i.e. 1 dès qu'il y en a une à 1, 0 s'il n'y en a pas.

[4] En vue d'une programmation plutôt que d'un calcul manuel.

Interprétation

Que signifie $R^m[x, y] = 1$? Qu'il y a un *chemin* de taille m entre x et y. La plus petite valeur de m qui satisfait cette relation correspond à un chemin élémentaire. Au-delà, d'autres valeurs sont possibles du fait d'autres chemins élémentaires. De plus, tout chemin élémentaire de taille m passant par un sommet appartenant à un circuit de taille k donne un chemin de taille m + k·n s'il parcourt le circuit n fois.

$R^m[x, x] = 1$ signifie que x est sur un *circuit* de taille m. La plus petite valeur de m satisfaisant cette relation correspond à un circuit élémentaire. Au-delà, tout circuit élémentaire de taille m passant par un sommet appartenant à un autre circuit de taille k permettra de constituer des circuits de taille n·m + p·k (n, p quelconques)...

Ainsi,
$R^3(b, b) = 1$ par le circuit élémentaire (b, f, d, b),
$R^3(a, d) = 1$ par (a, d, f, d), chemin non élémentaire, contenant le circuit (d, f, d).

Pour connaître seulement l'existence les chemins de taille 4 au plus,

- calculer $(((R + I)\cdot R + I)\cdot R + I)\cdot R + I$ demande $3n^3 + 4n^2$ opérations ;
- calculer ${}_0R_4 = ({}_0R_2)^2 = ({}_0R_1)^4 = (I+R)^4$ demande $2n^3 + n^2$ opérations, soit un gain d'au moins 33%.

8.5.3 Existence de chemins

Discussion

Si l'on veut savoir seulement s'il existe ou non des chemins de x à y dans un graphe G, ceci s'entend pour n'importe quelle taille possible.

Il faudrait donc pour un graphe donné calculer une matrice ${}_0C_\infty$ telle que ${}_0C_\infty[x, y]$ soit la réponse.

L'union de toutes les puissances de R possibles, de 0 à l'infini, ne serait pas exécutable en un temps fini.

Mais nous ne nous intéressons en général qu'aux *graphes finis*, à $|X| = n$ sommets.

Comme à tout chemin correspond au moins un chemin élémentaire, il n'existe de chemin de x à y que s'il existe un chemin élémentaire de x à y : ceux-ci étant sans répétition de sommets, ils sont tous fournis par la détermination des chemins de taille strictement inférieure à n. Ainsi, *pour un graphe à n sommets :*

$$ {}_0C_\infty = {}_0C_{n-1} $$

PROPOSITION 8.12 – La suite des relations ${}_0C_k$ est formée de relations de moins en moins fines, et toutes les ${}_0C_k$ deviennent identiques pour $k \geq n$.

PREUVE. La suite des relations ${}_0C_k$ est de moins en moins fine par définition : ${}_0C_{k+1} = {}_0C_k \cup R^{k+1}$ entraîne que ${}_0C_k \subseteq {}_0C_{k+1}$ (1).

Toutes les ${}_0C_k$ deviennent identiques pour $k \geq n$: en effet, tout « nouveau » chemin S de x à y de taille supérieure ou égale à n, comporte au moins n arcs et donc n + 1 sommets pris parmi n. Un sommet au moins est fatalement répété, et S comprend au moins un circuit. L'existence du chemin S de x à y a suppose donc l'existence d'un chemin élémentaire S', sous-chemin de S, de taille k'<n, d'où

$x\ {}_0C_k\ y \wedge k \geq n \rightarrow (\exists\ k' \leq n\ x\ {}_0C_{k'}\ y) \rightarrow (x\ {}_0C_n\ y)$ soit

${}_0C_k \wedge k \geq n \rightarrow {}_0C_k \subseteq {}_0C_n$

Or on a déjà $\forall\ k \geq n \quad {}_0C_n \subseteq {}_0C_k$ par (1),

donc $\forall\ k \geq n \quad {}_0C_k = {}_0C_n$ ■

Procédé

Pour décider de l'existence de chemins, entre tout sommet x et tout sommet y d'un graphe G à n sommets, calculer la matrice $[R \cup I]^m$, où m est la plus petite puissance de 2 supérieure ou égale à n.

→ Coût en $\lceil \log_2 n \rceil \cdot n^3$

n = 39 demanderait

- 39 produits de matrice par une approche purement multiplicative,
- 8 produits de matrice par la méthode de Legendre,
- mais 6 produits de matrices seulement pour calculer la puissance 64ème, égale à la puissance 39ème, au moyen d'environ 360 000 opérations ET et OU.

Tableau 8.4 Recherche de chemins par carrages successifs.

${}_0R_1$	a	b	c	d	e	f	g
a	1	o	o	1	1	o	o
b	o	1	1	o	o	1	o
c	o	o	1	o	o	o	o
d	o	1	o	1	o	1	o
e	1	o	o	o	1	o	o
f	o	o	o	1	o	1	1
g	o	o	o	o	o	o	1

⇒

${}_0R_2$	a	b	c	d	e	f	g
a	1	1	o	1	1	1	o
b	o	1	1	1	o	1	1
c	o	o	1	o	o	o	o
d	o	1	1	1	o	1	1
e	1	o	o	1	1	o	o
f	o	1	o	1	o	1	1
g	o	o	o	o	o	o	1

${}_0R_4$	a	b	c	d	e	f	g
a	1	1	1	1	1	1	1
b	o	1	1	1	o	1	1
c	o	o	1	o	o	o	o
d	o	1	1	1	o	1	1
e	1	1	1	1	1	1	1
f	o	1	1	1	o	1	1
g	o	o	o	o	o	o	1

=

${}_0R_8$	a	b	c	d	e	f	g
a	1	1	1	1	1	1	1
b	o	1	1	1	o	1	1
c	o	o	1	o	o	o	o
d	o	1	1	1	o	1	1
e	1	1	1	1	1	1	1
f	o	1	1	1	o	1	1
g	o	o	o	o	o	o	1

Si nous reprenons notre exemple (graphe G_2) :

- il y a 7 sommets, donc on calcule ${}_0R_8$ à partir de ${}_0R_1$, ${}_0R_2$, ${}_0R_4$ (tab. 8.4),
- de fait, ${}_0R_4 = {}_0R_8$, car
  - il n'y a qu'un seul chemin *élémentaire* de taille 5, et un chemin *élémentaire* de taille 4 a les mêmes extrémités,
  - il n'y a pas de chemin *élémentaire* de taille 6 (fig. 8.3).

8.5.4 Chemins optimaux

Chemins de taille minimale

Il existe un chemin de *taille 1* entre x et y s'il existe un arc de x à y .

Il existe un chemin de *taille m* entre x et y s'il existe un z tel qu'il existe un chemin de taille k entre x et z, et un chemin de taille m – k entre z et y.

Il existe enfin un chemin de *taille minimale m* entre x et y s'il existe un chemin de taille m entre x et y, et si tous les autres chemins entre x et y sont de taille m au moins.

Ce chemin de *taille minimale m* entre x et y est *le plus simple* des chemins reliant x à y.

Les calculs matriciels précédents restent valables à condition :

- de traiter des matrices d'entiers et non plus de booléens,
- de combiner par addition les (tailles de) sous-chemins successifs,
- d'utiliser l'opérateur *min* pour trouver la taille associée au plus court chemin de 2 ou plusieurs chemins de mêmes extrémités,
- de coder l'absence d'arc par + ∞, l'absence d'arc ou de chemin étant assimilée à un chemin de taille infinie (ou impossible).

Ces calculs se stabilisent (tab. 8.5) car :

PROPOSITION 8.13 – Tout chemin de taille minimale est élémentaire (par l'absurde).

Tableau 8.5 Recherche de chemins de taille minimale.

$_0R_1$	a	b	c	d	e	f	g
a	0	∞	∞	1	1	∞	∞
b	∞	0	1	∞	∞	1	∞
c	∞	∞	0	∞	∞	∞	∞
d	∞	1	∞	0	∞	1	∞
e	1	∞	∞	∞	0	∞	∞
f	∞	∞	∞	1	∞	0	1
g	∞	∞	∞	∞	∞	∞	0

⇒

$_0R_8$	a	b	c	d	e	f	g
a	0	2	3	1	2	1	3
b	∞	0	1	2	∞	1	2
c	∞	∞	0	∞	∞	∞	∞
d	∞	∞	2	0	∞	1	2
e	1	3	4	1	0	3	4
f	∞	2	3	1	∞	0	1
g	∞	∞	∞	∞	∞	∞	0

x	a	b	c	d	e	f	g	
$_0R_2$ (d,x)	∞	*1*	2	0	∞	1	2	
$_0R_2$ (x,c)	∞	*1*	0	2	∞	∞	∞	min
+	∞	*2*	2	2	∞	∞	∞	⇒*2*

calcul d'un élément $_0R_4$ (d,c) :

$_0R_2$ (d,b) + $_0R_2$ (b,c) = 2

⇒ $_0R_4$ (d,c) (<) = 2

Chemins de longueur minimale

Supposons maintenant les *arcs valués par des distances.*

A un chemin, possédant déjà une *taille* (nombre de ses arcs = indice de complexité), nous associons de plus une ***longueur*** ou *somme des distances* attachées à ses arcs (indice de *coût*).

La matrice initiale est formée de valeurs non-négatives, les calculs sont les mêmes que ci-dessus.

On opère la combinaison des valeurs

- de *deux sous-chemins bout à bout par* +,
- de *deux chemins de mêmes extrémités par min*.

Les puissances à calculer restent bornées, car dès lors que tout circuit a une longueur strictement positive, les chemins de longueur minimale sont tous des chemins élémentaires (par l'absurde).

NOTE. On obtient de même les chemins de *coût* minimal (coût remplaçant distance et longueur).

Chemins de taille ou de longueur maximale

Ici, on ajoute encore la valeur de deux chemins mis bout à bout, mais on combine la valeur de deux chemins de mêmes extrémités par max : on retient la taille (resp. longueur) du plus long.

Les puissances à calculer restent bornées, aux conditions ci-après, assurant que toute solution est un chemin élémentaire.

PROPOSITION 8.14 – La recherche de chemins de taille (resp. longueur) maximale n'est convergente que dans les graphes sans boucle ni circuit.

Les chemins de taille maximale apparaissent entre autres dans le calcul des *fonctions ordinales*, en vue notamment du tracé automatique des graphes (exercice 8.11.2, sect. 10.6).

Pour les chemins de longueur maximale,

DÉFINITION 8.6 – Est dit ***chemin critique*** entre deux points tout chemin de longueur maximale. Les autres chemins de même extrémités de longueur inférieure, sont dits ***chemins sous-critiques.***

NOTES

- S'il existe un seul chemin critique de longueur maximale, il est commode d'organiser le tracé d'un graphe sans circuit autour de ce chemin critique pris comme axe.
- En matière d'ordonnancement, on peut modéliser un projet par un graphe, chaque point symbolisant un *événement* (ou date) du projet, et chaque arc (x, y) une *tâche* à réaliser, de durée connue (assimilée à une longueur de temps) commençant au plus tôt en x et se terminant au plus tard en y. On rend compte ainsi des précédences à respecter comme des simultanéités possibles (cf. problème 8.12.4). En distinguant deux événements : un *début*, antérieur à toutes les tâches, et une *fin*, postérieure à toutes les tâches, la durée du projet est celle du/des chemin(s) critique(s) du début à la fin. Pour *accélérer un projet*, on affectera des ressources à l'accélération des tâches (appartenant au(x) chemin(s)) critiques. Pour *rendre le projet plus économique*, on pourra réduire les ressources affectées aux tâches non critiques. Ces stratégies doivent être appliquées/peuvent être combinées pas à pas, en étant contrôlées par réévaluation.

Tableau 8.6 Recherche matricielle de chemins optimaux.

chemins	minimale	maximale
de taille	chemin le plus simple	
de longueur	chemin le plus court	chemins critiques
combinaison des chemins bout à bout des chemins de mêmes extrémités	+ min	+ max
calcul des puissances de matrice jusqu'à stabilisation ou dépassement	toute solution est élémentaire	toute solution est élémentaire, le résultat n'étant défini que pour des graphes sans circuit

Autres problèmes de chemins optimaux

De nombreux autres problèmes de chemins optimaux (de capacité maximale, de fiabilité maximale, ...) ont donné naissance à un foisonnement d'algorithmes similaires. La théorie des dioïdes (chap. 14) a précisément été créée pour unifier l'approche de cette famille de problèmes.

8.6 MÉTHODE LOCALE 1 : RECHERCHE PAR NIVEAUX

8.6.1 Principe

Considérons le graphe donné au tableau 8.7.

Comment aller de e à g ?

Une solution consiste à partir de e, puis dresser la liste des sommets que l'on peut atteindre en une étape, puis la liste des sommets que l'on peut atteindre en 2 étapes, 3, 4, 5...

Ce procédé est-il fini ?

Si on accepte de repasser plusieurs fois au même sommet, on peut tourner inexorablement en rond.

Cependant, s'il existe un chemin de e à g, alors il existe un chemin *élémentaire* de e à g ; recherchons ce dernier, solution allégée du problème posé.

Si on refuse de repasser deux fois par le même sommet,

- *on part de e,*
- *on dresse la liste des sommets autres que e que l'on peut atteindre en une étape,*
- *puis la liste des sommets non encore rencontrés qu'on peut atteindre en 2 étapes, 3, 4, 5...*
  - *si on atteint une liste de successeurs vide avant d'avoir atteint le but (g) c'est qu'il n'est pas accessible ;*
  - *sinon, g est nécessairement atteint en moins de N étapes.*

Tableau 8.7 Graphe.

Sommet	Successeurs
a	d, e
b	c, f
c	∅
d	b, f
e	a
f	d, g
g	h, i ...

Tableau 8.8 Exploration par niveaux.

Niveau	Successeurs
0	e
1	a
2	d, e
3	a, b, f
4	c, f, d, g

8.6.2 Existence d'un chemin

Elle est décidable par la fonction booléenne ci-après.

test RechercheParNiveaux (G : graphe ; sommetInitial, but : sommet)

« test d'existence d'un chemin de *sommetInitial* à *but* dans *G* »
SousNiveau,Visités: ensemble de sommets ;
Liste : file de sommets ;
x : sommet ;

Liste, Visités ←{sommetInitial}

tantque Liste ≠ ∅

```
SousNiveau← G.succ(premier(Liste)) – Visités
si but ∈ SousNiveau rend vrai ;
Liste←reste(Liste)      « Liste décapitée »
AjoutEnQueue(Liste, SousNiveau)
Visités←Visités ∪ SousNiveau
```

rend faux ;

fin RechercheParNiveaux

8.6.3 Enonciation de chemins

Le procédé est un peu plus élaboré. Le test d'existence recherche le but parmi les descendants du sommet initial, dans un *faisceau* des chemins issus du sommet initial, dont le développement s'arrête lorsque le but est atteint. Il reste à extraire du faisceau le(s) chemin(s) effectif(s), dont les sommets doivent être aussi ancêtres du but. Pour cela, partant du but, on remonte jusqu'au niveau 0 en ne gardant dans chaque niveau k que les prédécesseurs des sommets gardés du niveau k + 1.

Tableau 8.9 Recherche d'un chemin par niveaux de e à g, avec énonciation du chemin.

Niveau	Successeurs nouveaux / possibles	Prédécesseurs (à retenir)
0	e	$e \in \Gamma^{-1} a$
1	a	$a \in \Gamma^{-1} d$
2	d	$d \in \Gamma^{-1} f$
3	b, f	$f \in \Gamma^{-1} g$
4	~~e,~~ g	

8.6.4 Chemins de taille minimale (resp. bornée)

Obtention

Le procédé par niveaux s'arrête à la découverte du *premier chemin élémentaire trouvé,* qui est *de taille minimale* puisque l'exploration est menée par tailles croissantes, taille après taille.

En limitant le nombre de niveaux, le procédé recherche un chemin de taille bornée (inexistant si la borne est insuffisante).

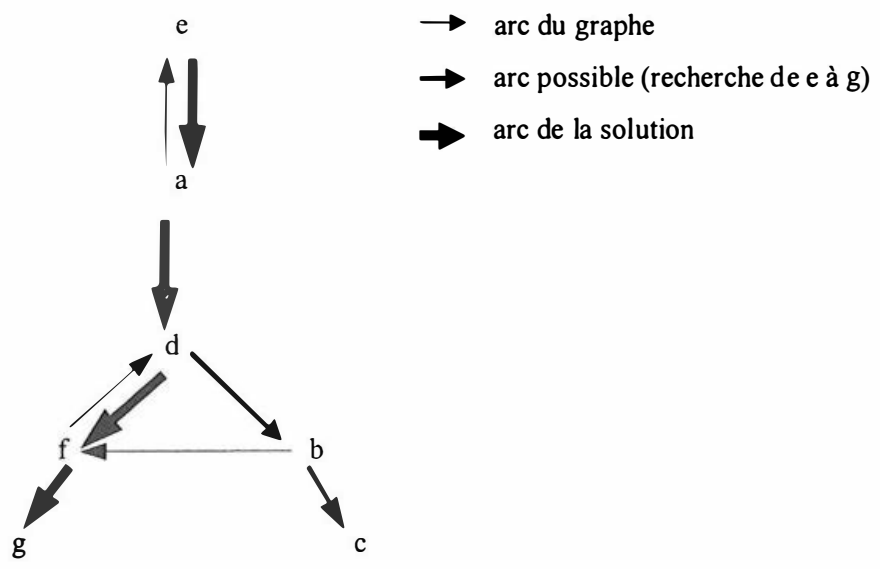


Fig. 8.6 Recherche de chemins par niveaux de e à g.

Complexité

Soit k la *sortance*[5] *moyenne* d'un sommet du graphe ; quand un niveau contient p sommets, le niveau suivant contient un nombre de sommets de l'ordre de k·p. De ce point de vue, le nombre de sommets traités varie comme $1 + k + k^2 + \ldots + k^{dernierNiveau} = O(k^{dernierNiveau})$ dès lors que $k > 1$, tout en restant au plus égal à N.

La mémoire occupée sera du même ordre du seul fait de la gestion de l'ensemble des visités.

Ainsi, *aux N très élevés, le procédé est considéré comme vorace en place,* quand l'encombrement en $O(k^{dernierNiveau})$, petit devant N, reste excessif vis-à-vis des ressources disponibles.

Supposons que l'on puisse stocker explicitement S sommets.

La méthode par niveau s'applique aisément aux graphes ayant moins de S/2 sommets. Au-delà, pour les graphes à N sommets de sortance moyenne k, le procédé devra se borner à la recherche des chemins de taille inférieure à $\log_k S$.

Ainsi, pour S =10 000,

- pour k = 2 on peut atteindre les niveaux 12/13,
- mais pour k = 5 on devra se limiter aux niveaux 5/6,
- et pour k = 20 (jeu d'échecs) on devra se limiter au niveau 3.

En réalité, la valeur de k à prendre est le nombre *moyen* de successeurs *nouveaux* : l'analyse précédente, valable pour les graphes sans circuit, est pessimiste pour les graphes à circuit.

[5] Ou *nombre de successeurs.*

1^8	2^7	3^6	4	5	6	7	8^9	9^8	0^9
A^7	Z^6	E^5	R^4	T^3	Y^4	U^5	I^6	O^7	P^8
Q^6	S^7	D^6	F^3	G^2	H^5	J^6	K^7	L^8	M^9
W^5	X^4	C^3	V^2	B^1	N^0				

de N à M : niveau en exposant, chemins NBGTYUIO(P ou L)M.

Fig. 8.7 Recherche par niveaux dans un labyrinthe.

Dans le labyrinthe ci-dessus, $k \leq 4$, et l'on voit que le procédé trouve une solution simple, en laissant inexplorés les sommets trop éloignés du point de départ.

8.6.5 Chemins de longueur minimale

Comme dans le paragraphe 8.5.6, on s'intéresse maintenant à des *graphes valués,* dont chaque arc est porteur d'une valeur positive spécifique, assimilée à une longueur (ou à un coût), et la longueur (resp. le coût) d'un chemin est la somme des valeurs associées à ses arcs.

Quand on peut prouver que le chemin de longueur minimale est de taille minimale, on adapte la méthode par niveaux. C'est la cas pour un graphe complet dont la valuation satisfait l'inégalité du triangle.

Sinon, on utilise la méthode du meilleur d'abord.

8.7 MÉTHODE LOCALE 2 : MÉTHODE DU MEILLEUR D'ABORD

8.7.1 Principe

Quand on cherche un chemin de longueur minimale sans pouvoir compter sur des propriétés particulières, on utilise une méthode apparentée à la méthode par niveaux, qui est aussi un *algorithme-glouton,* c'est-à-dire qui procède sans remise en cause.

En progressant du sommet initial vers le but, la méthode distingue :

- les *sommets traités*, pour lesquels on connaît un chemin minimal depuis le sommet initial, et qui ne participent plus au traitement (mais pourront contribuer à l'édition finale),
- les *sommets de la frange*, évoqués au moins une fois, pour lesquels on connaît un chemin et son coût ; la frange forme une zone mobile qui généralise la notion de niveau.

Selon Bellman (1964), *tout sous-chemin d'un chemin optimal est optimal.*

Un chemin optimal à n + 1 arcs n'est alors qu'un chemin formé d'un chemin optimal à n arcs, « prolongé au mieux » d'un arc.

C'est de plus un chemin élémentaire, dès lors que tous les coûts sont positifs, et les sommets déjà visités n'ont pas à être réemployés.

8.7.2 Procédé Dijkstra-Moore

On utilise le procédé suivant.

Au départ, on installe l'origine o dans la frange sous la forme d'un triplet (o, 0, o) ; l'ensemble des sommets traités est vide.

Ensuite, l'algorithme repère dans la frange le point d'arrivée p du plus court chemin issu du sommet initial (s'il n'y en a pas, le procédé a échoué : le point d'arrivée n'est pas accessible).

Aussi longtemps que ce sommet p n'est pas le but,

- *l'algorithme envisage tous les successeurs x de ce point p :*
  - *si le successeur x a déjà été traité, on l'ignore ;*
  - *si le successeur x n'a jamais été rencontré, il entre dans la frange sous une forme (x, l, p), où l désigne la longueur de chemin trouvée, et p son prédécesseur immédiat sur ce chemin ;*
  - *sinon, le successeur x est déjà dans la frange sous une forme (x, λ, π) ; soit (x, l, p) la nouvelle possibilité : si $\lambda < l$, on garde l'ancienne solution, sinon on lui substitue (x, l, p) ;*
- *quand tous les successeurs de p ont été envisagés,*
  - *l'actuel p est retiré de la frange et ajouté à la liste des sommets traités ;*
  - *un nouveau point d'arrivée p' du plus court chemin allant du sommet initial à un point de la frange est défini.*

Quand le but est atteint, le coût du chemin trouvé est immédiatement connu ; pour son énonciation, on remonte du but à l'origine o par la chaîne des prédécesseurs immédiats.

La connaissance de l'ensemble du graphe n'est pas nécessaire, dès lors que les successeurs d'un sommet quelconque de la frange peuvent toujours être produits, par exemple par un procédé de calcul ou par un système de règles.

8.7.3 Exemple

Une carte donne les « distances » du tableau 8.10.

Quel est le *plus court chemin* allant de A à D ?

On applique le procédé ci-après.

Les états successifs des listes Traités et Marqués sont donnés dans le tableau 8.11.

E apparaît tôt dans Marqués du fait d'une liaison AE directe, mais n'est traité qu'après C, plus près en termes de distance mais auquel on accède plus tard par un chemin de taille 2.

Tableau 8.10 Distances.

A				
7	B			
/	10	C		
/	/	7	D	
20	/	7	10	E

procédé RechercheParValeur (graphe G ; sommetInitial, but : sommet)

type arc = fiche (arrivée : sommet ; valeur : positif ; prédécesseur : sommet)

« le prédécesseur est noté pour faciliter l'édition finale »

Marqués : liste d'arcs « frange mobile»
Traités : liste d'arcs « partie figée »

Traités←{}
Marqués←arc(sommetInitial,0,sommetInitial)

tantQue Marqués≠{}

x←arrivée (tête (Marqués))
valx←valeur (tête (Marqués))
si but = x *« succès : résultat explicité par chaînage dans Traités »*
Traités←Traités ∪ tête (Marqués)
Marqués←reste(Marqués)
pour chaque y∈G.succ(x)–Traités faire
Marqués←Insertion(Marqués, arc (y, valx+G.valeur(x,y), x)

« où Insertion
- insère dans Marqués tout arc (y, v, x) dans l'ordre des v croissants
- au-delà d'un premier (y, v, x), élimine tout triplet (y, v', x) de valeur v' strictement supérieure à v

»

« échec : but n'est pas accessible »

fin RechercheParValeur

Tableau 8.11 Etapes d'une recherche « le meilleur d'abord » pour le graphe 8.10.

Traités	Examiné	Marqués
	(A 0 A)	(B 7 A) (E 20 A)
(A 0 A)	(B 7 A)	(C 17 B) (E 20 A)
(A 0 A)(B 7 A)	(C 17 B)	(E 20 A) (D 24 C)~~(E 24 C)~~
(A 0 A)(B 7 A)(C 17 B)	(E 20 A)	(D 24 C) ~~(D 30 E)~~
(A 0 A)(B 7 A)(C 17 B)(E 20 A)	(D 24 C)	

A la fin, le chemin s'obtiendra par chaînage en remontant de prédécesseur en prédécesseur :

(D 24 C) ⇒ (D 24 (C 17 B))⇒ (D 24 (C 17 (B 7 A))),
et longueur(ABCD) = 24 < longueur(AED) = 30
mais taille(ABCD) = 3 > taille(AED) = 2

Le chemin le plus simple (AED) n'est donc pas ici le plus court (ABCD).

8.8 MÉTHODE LOCALE 3 : RECHERCHE EN PROFONDEUR

La méthode par niveaux, qui rappelle une propagation d'onde, suggère une sorte de parallélisme dans les recherches, qui n'est pas toujours réalisable.

Au contraire, une personne dans un labyrinthe procédera généralement par *essais et erreurs,* mais peut le faire de façon systématique ; cette méthode, plutôt myope, est assez médiocre mais moins coûteuse en mémoire.

8.8.1 Principe

Au point de départ du labyrinthe de la figure 8.8 appliquons le procédé suivant :

- étant en un point x,
  - j'examine les successeurs possibles non encore rencontrés ;
  - si j'ai atteint le/un but, « réussite » ;
  - sinon, parmi les successeurs possibles ne faisant pas déjà partie du chemin parcouru, je choisis « le premier encore inexploré » (par exemple, avec une priorité Est / Nord / Ouest / Sud) comme nouvel x, et je progresse ainsi aussi longtemps que possible ;
  - si j'arrive à une impasse (point sans successeur inexploré) ou à un lieu dont toutes les issues ont été explorées négativement, je retourne au dernier point quitté et j'y condamne mon dernier choix ; cette régression se poursuit si cette condamnation transforme ce point en impasse, sinon je traite le premier cas restant ;
- si je me retrouve au point de départ sans issue inexplorée « échec ».

1	2	3	4	5	6	7	8	9	0
A	Z	E	R	T^3	Y^4	U^5	I^6	O^7	P^8
Q	S	D	F	G^2	H	J	K	L	M^9
W	X	C	V	B^1	N^0				

Fig. 8.8 Recherche en profondeur N-M (priorité E N O S).

La méthode comporte un *aspect arbitraire : la sélection à un endroit donné, s'il en reste plus d'une, de la première issue inexplorée que l'on va explorer.*

Ici, le système de priorité Est / Nord / Sud / Ouest se révèle favorable au cas N–M, car détecter M comme successeur de P évite d'explorer l'impasse 0-9-8-7-6-5-4, mais ce n'est pas toujours vrai.

Supposons que je cherche à aller de N vers A.

La recherche démarre de même, « en aveugle ».

Arrivé en P, je dois essayer 0 /O/M : il n'y a plus capture par le but, et je passe en 0 ; de là, les issues inexplorées sont à l'Ouest, et je vais en 9, 8, 7, 6, 5, 4 pour trouver que 4 est sans issue : donc 5, 6, 7, 8 étaient des impasses. En 9, O faisant déjà partie du chemin n'est pas envisageable comme alternative à 8. 9 est donc une impasse, donc 0 qui ne peut me ramener à P comme issue, et y revient par échec : comme en P je venais de O, seul M est explorable au pas 23, qui ne peut mener qu'à L (24) et L au mur. Donc M est une impasse, et P et O, et je retourne en I ; je repars en K (pas 29), qui mène à J, H et au mur ; donc J, K, I, U, Y sont des impasses : revenus en T, je tente R, E, 3, 2, 1, A, et je trouve A au pas 43.

Les contre-marches étant effacées de la pile de manœuvre, il reste un chemin NBGTRE321A, de taille 9 (supérieure à la taille 7 du chemin optimal NBVCXWQA).

Ce deuxième cas souligne que, telle quelle,

l'exploration en profondeur trouvera une solution s'il y en a une,
sans garantie de vitesse ni de qualité,
du fait d'errances possibles dans des portions inadéquates
du labyrinthe ou du graphe.

1^{42}	2^{41}	3^{40}	4^{15}	5^{14}	6^{13}	7^{12}	8^{11}	9^{10}	0^{9}
A^{43}	Z	E^{39}	R^{38}	T^{3}	Y^{4}	U^{5}	I^{6}	O^{7}	P^{8}
Q	S	D	F	G^{2}	H^{31}	J^{30}	K^{29}	L^{24}	M^{23}
W	X	C	V	B^{1}	N^{0}				

Fig. 8.9 Recherche en profondeur N-A (priorité E N O S).

8.8.2 Existence et énonciation

La recherche en profondeur confond les deux problèmes en prouvant l'existence d'un chemin entre x et y par la construction (par essais et erreurs successifs) de l'énoncé de l'un des chemins élémentaires possibles de x à y.

8.8.3 Complexité et performance

La méthode n'est pas très rapide ni de très grande qualité, mais pour les grands graphes elle est peu gourmande en mémoire, comparée à la méthode par niveaux. En effet, si k est la sortance moyenne d'un sommet, la recherche en profondeur atteint le niveau n en stockant k·n sommets.

Pour les grands graphes, supposons que l'on puisse stocker explicitement S sommets au plus.

Au-delà, pour les graphes à n sommets de sortance moyenne k, le procédé permettra la recherche des chemins de taille S/k au plus.

Ainsi, pour S = 10 000,

- pour k = 2 on peut atteindre les niveaux 5000,
- pour k = 5 on atteindra les niveaux 2000,
- et pour k = 20 on pourra encore atteindre le niveau 500.

Comme précédemment, la valeur de k à considérer réellement est le *nombre moyen de successeurs nouveaux* : l'analyse précédente est pessimiste pour les graphes à circuit.

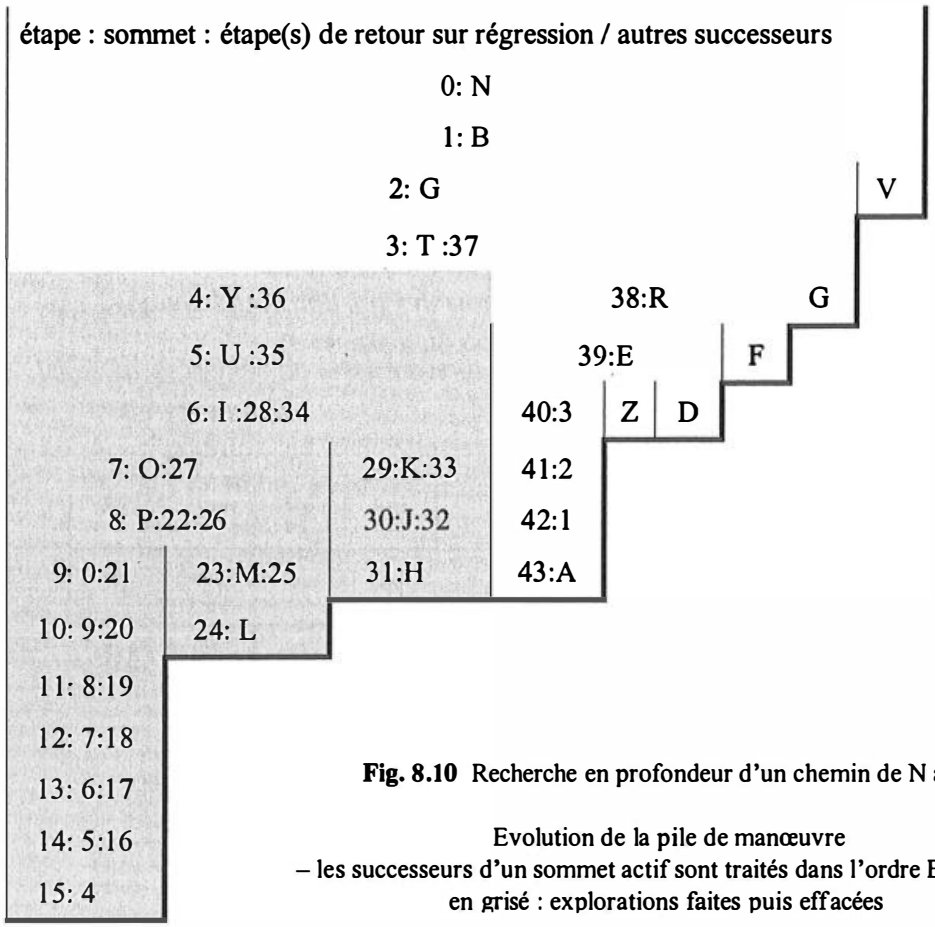


Fig. 8.10 Recherche en profondeur d'un chemin de N à A.

Evolution de la pile de manœuvre
– les successeurs d'un sommet actif sont traités dans l'ordre E/N/O/S –
en grisé : explorations faites puis effacées

8.8.4 Heuristiques

Cette méthode est en soi économe en place mais médiocre dans ses résultats et parfois lente du fait de sa myopie et de ses errances.

Lors de l'adaptation du procédé générique à un problème connexe,
on utilisera le plus souvent les contraintes propres au problème
comme principe de pilotage de la recherche en profondeur,
accroissant ainsi pertinence de la réponse et performance de l'algorithme.

Chemin de taille bornée, de taille minimale

Le procédé de recherche en profondeur s'adapte simplement à la recherche d'un chemin élémentaire de taille bornée, en traitant en impasse le fait que la taille du sous-chemin déjà développé atteint la borne prescrite.

Pour la recherche d'un chemin de taille minimale, une première idée est de modifier le procédé pour que la recherche se poursuive au-delà de la première solution, de façon à énumérer tous les chemins élémentaires attachés au couple de points à relier.

Cependant il serait stérile d'examiner des solutions moins intéressantes que celles déjà rencontrées, si la meilleure solution n'est pas la dernière.

On procèdera donc le plus souvent par *amélioration progressive,* la taille du meilleur chemin déjà trouvé servant de borne à la poursuite de la recherche.

Recherche d'un chemin de *taille* minimale _____

- prendre comme borne initiale de la taille du chemin, le nombre de sommets ;
- énumérer par une recherche en profondeur *tous les chemins de taille strictement inférieure à la borne, égale à chaque instant à la taille du dernier chemin trouvé* ;
- on forme ainsi une suite de chemins élémentaires de taille strictement décroissante : *... le chemin de taille minimale est le dernier trouvé.*

fin recherche d'un chemin de taille minimale _____

8.8.5 Chemin de longueur minimale

Le procédé est calqué sur le précédent.

Recherche d'un chemin de *longueur* minimale _____

- prendre comme borne initiale de la longueur du chemin, un nombre majorant largement la longueur espérée ;
- énumérer par une recherche en profondeur *tous les chemins de longueur strictement inférieure à la borne, égale à chaque instant à la longueur du dernier chemin trouvé* en commençant à chaque niveau par le successeur atteint par l'arc le plus court ;
- on forme ainsi une suite de chemins élémentaires de longueur strictement décroissante : *le chemin de longueur minimale est le dernier trouvé.*

fin Recherche de chemin de longueur minimale _____

Dans les cas où l'on cherche une *réponse de qualité suffisante dans un délai bref*, on peut utiliser une version *interruptible* (par exemple, à chaque nouvelle solution) ou *à durée limitée* de ce procédé d'***optimisation progressive.***

8.9 PROBLÈMES HAMILTONIENS

DÉFINITION 8.7 – En matière de graphes, on appelle ***problèmes hamiltoniens*** *les problèmes relatifs aux circuits et chemins passant une fois et une seule par chaque sommet d'un graphe.*

COROLLAIRE 8.15 – Pour un graphe à N sommets, un chemin hamiltonien est un chemin élémentaire de taille N – 1, un circuit hamiltonien est un circuit élémentaire de taille N.

PROPOSITION 8.16 (Redei) – Si un graphe est complet, il possède un chemin hamiltonien.

PREUVE. On suppose un graphe $G = \langle X, U \rangle$ tel que pour chaque couple de sommets (x y) on ait soit $(x\ y) \in U$ soit $(y\ x) \in U$.

Tout arc peut être considéré comme un chemin élémentaire de taille 1.

Soit $(x_0, x_2, \dots x_k, \dots x_n)$ un chemin élémentaire de taille $n < N$. Soit x un sommet de X n'appartenant pas à ce chemin. Il existe un arc entre x et chacun des x_i. En raisonnant sur leur orientation, du premier au dernier, de trois choses l'une :

- ou $(x\ x_0)$ est un arc, et $(x, x_0, x_2, \dots x_k, \dots x_n)$ est un chemin élémentaire de taille n + 1,
- ou il existe x_k tel que $(x_k\ x)$ et $(x\ x_{k+1})$ sont des arcs, et $(x, x_0, x_2, \dots x_k, x, x_{k+1} \dots x_n)$ est un chemin élementaire de taille n +1 ;

- ou finalement $(x\ x_n)$ n'est pas un arc, alors $(x_n\ x)$ en est un, et $(x_0, x_2, \ldots x_k, \ldots x_n, x)$ est un chemin élementaire de taille $n + 1$.

Par récurrence, on voit donc qu'un graphe complet à N sommets comporte au moins un chemin élémentaire pour chaque taille possible.

Il comprend notamment un chemin élémentaire de taille $N - 1$, le chemin hamiltonien cherché. ■

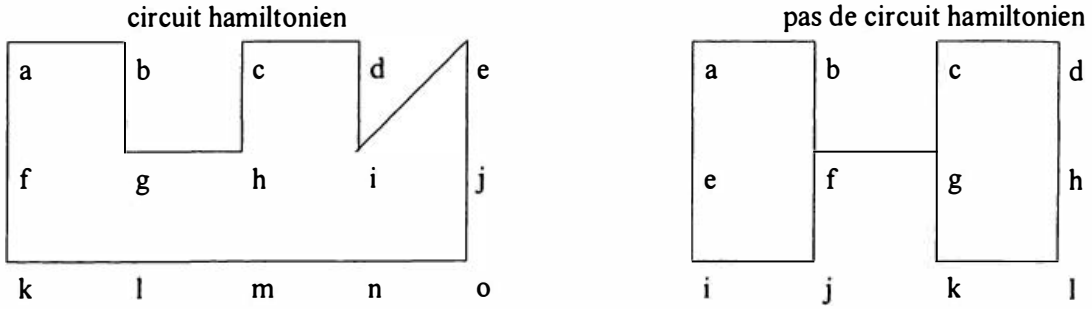


Fig. 8.11 Graphes avec et sans circuit hamiltonien.

EXEMPLES. Font partie de cette classe :

- le voyage fermé autour du monde de W.R. Hamilton : on suppose 20 villes sur le globe terrestre, aux sommets d'un dodécaèdre régulier (20 sommets, 12 faces pentagonales) : on cherche un itinéraire qui, partant d'une ville, passe une fois et une seule par chacune des 19 autres villes, puis retombe sur la ville de départ ;
- la course du cavalier sur l'échiquier (Euler (1766), de Moivre, Vandermonde...) ;
- le *problème du représentant de commerce* (*traveling salesman problem*) qui désire faire la tournée la moins longue possible, en passant une fois et une seule par les sites d'une liste ;
- le problème des courriers/coursiers entre divers établissements d'une même entreprise ;
- le problème d'une navette entre différents sites ;
- le problème des livres de S. Johnson (1954) : comment une même personne doit s'organiser pour imprimer et relier n livres différents dans un temps minimum ;
- divers problèmes de codage ;
- certains problèmes de tournoi.

(Voir aussi sect. 10.9).

8.10 PROBLÈMES EULÉRIENS

DÉFINITION 8.8 – En matière de graphes, on dit ***problèmes eulériens*** tous les problèmes relatifs aux chemins et circuits passant une fois et une seule par chaque arc d'un graphe.

PROPOSITION 8.17 – Tout graphe connexe dont les sommets sont de degré pair admet un cycle eulérien.

EXEMPLES. Font partie de cette classe :

- le problème des *ponts de Königsberg* : pouvait-on, au cours d'une promenade dans Königsberg[6], passer une fois et une seule par chacun des 7 ponts de la Pregel ? (Euler, 1736) [BERGE 58] ;
- le problème du *facteur chinois,* dont la tournée passe par chaque segment de rue une fois et une seule ;
- le problème des *tournées de bennes à ordures* ;
- le générateur de poèmes de de Werra ;
- la question des figures traçables d'un seul trait (ou à main levée) .

(Voir aussi sect. 10.10).

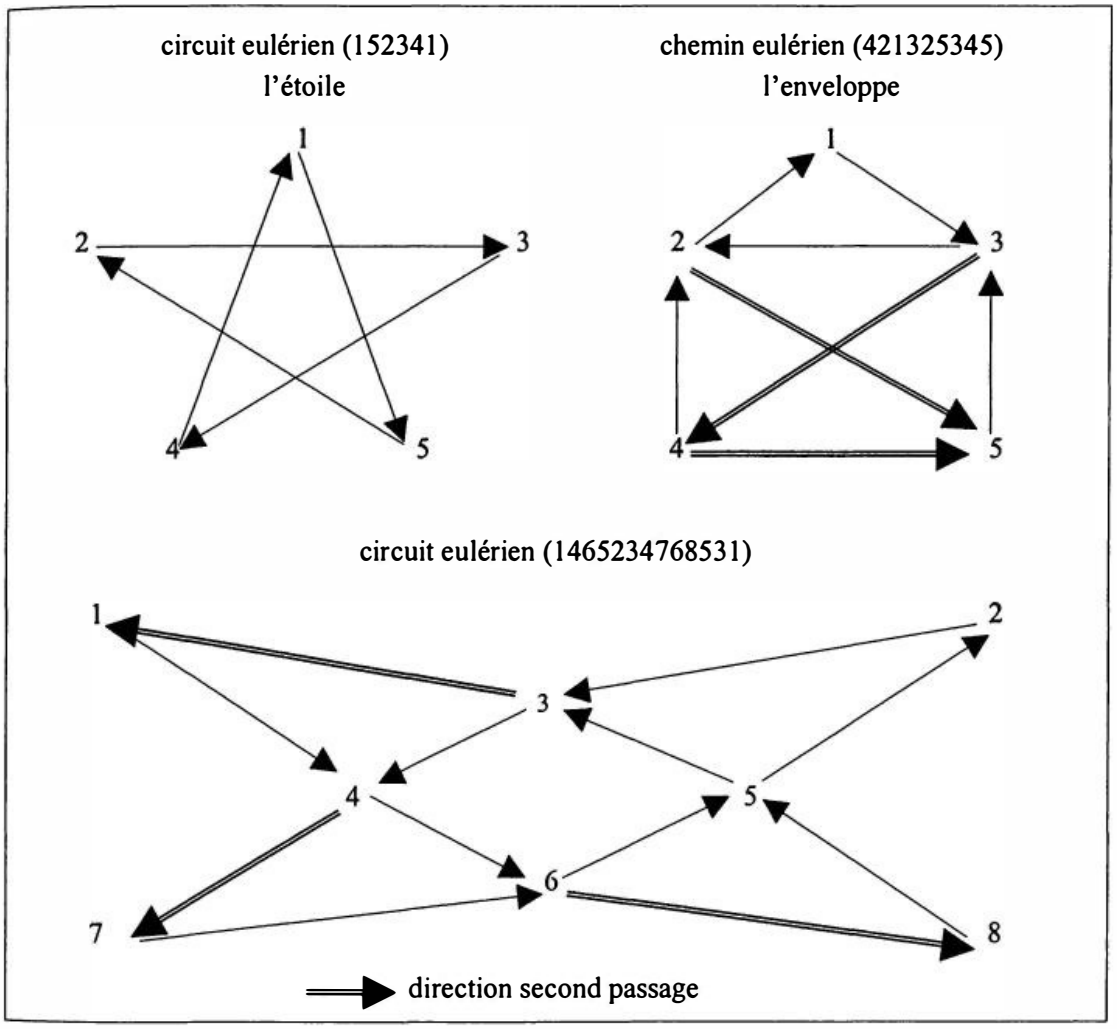


Fig. 8.12 Circuits et chemins eulériens.

[6] Maintenant Kaliningrad.

8.11 EXERCICES

8.11.0 (Huxtable, 1961)

On considère un programme Π formé d'un ensemble P de procédures p, et on considère le graphe G = (P,U) ainsi défini : il existe un arc (p1,p2) dans G ssi la procédure p1 appelle la procédure p2 dans le programme Π.

Expliciter la signification en termes de procédures et de programme des chemins, puis des boucles et des circuits de G. Quelle est la signification du circuit élémentaire de taille max passant par un point ?

8.11.1

Montrer qu'une relation transitive asymétrique possède un *graphe sans boucle ni circuit.*

Application

Ordonnancement de projet : on considère un projet défini par une liste d'éléments (t, d, t_1, t_2, t_3, ... t_n), qui fixent pour une tâche t sa durée d et la liste t_1, t_2, t_3, ... t_n des tâches qui doivent être terminées avant que t commence ; abstraction faite des durées, donner un *critère de validation globale des données* s'appuyant sur une *relation de précédence entre tâches.*

8.11.2 Fonction ordinale (Kaufmann)

On considère un graphe G = <X, U> et on définit sur ce graphe une *fonction ordinale*

$$f(x) = (\textit{si } \Gamma x = \varnothing \textit{ alors } 0 \textit{ sinon } 1 + \max(\{f(y) \mid y \in \Gamma x\})$$

Interpréter f(x) en temps que taille de chemin. Montrer que cette fonction n'existe que si le graphe G est *sans boucle ni circuit.*

On dira des sommets *de même niveau* s'ils ont même fonction ordinale.

Vérifier que l'existence d'une telle fonction pour un graphe G facilité son tracé si, partant d'une suite de droites parallèles numérotées successivement de 0 à n, on s'impose de représenter les sommets de niveau k sur la droite de rang k.

Montrer qu'on peut utiliser de même une fonction g_G déduite de f_G en permutant les rôles des sommets initiaux et terminaux.

8.11.3 Permutations

Soit une suite de n objets distincts. On appelle permutation une transformation p notée (i_1, i_2, ... i_n) où les i_1, i_2, ... i_n sont des entiers de 1 à n tous distincts, et telle que :

$$p\,(x_1, x_2, \ldots x_n) = (x_{i1}, x_{i2}, \ldots x_{in})$$

i.e. telle que l'objet de rang k dans la suite initiale devienne l'objet de rang i_k dans la suite permutée.

A chacune de ces permutations, on associe un graphe à n sommets.

Montrer que ce graphe n'est formé que de boucles et de circuits disjoints (appelées ***orbes***).

En appliquant ces considérations à la permutation (1 2 6 5 4 7 3), justifier la transcription (1 2 6 5 4 7 3) = (3 6 7)(4 5).

Par récurrence sur le nombre de circuits, montrer que pour toute permutation p il existe un plus petit entier k tel que la puissance de composition p^k soit la permutation identique.

En conclure que pour un ensemble à 12 objets, il existe une permutation dont seules les puissances multiples de 60 sont neutres.

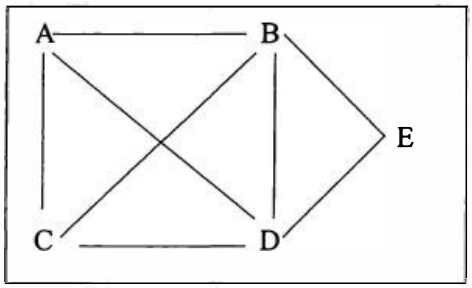
8.11.4 (R.Bellman)

Montrer que

- tout sous-chemin d'un chemin de taille minimale est de taille minimale ;
- tout sous-chemin d'un chemin de longueur minimale est de longueur minimale ;
- tout sous-chemin d'un chemin critique est critique.

8.11.5 (Prather)

On considère la carte ci-contre.



Un VRP peut-il, partant d'une ville, passer par chaque autre *ville* une fois et une seule, et revenir à la ville initiale ?

Ce schéma routier peut-il être tracé d'un seul trait, sans lever le crayon ni tracer 2 fois la même arête ?

Un facteur, un inspecteur peut-il parcourir chaque *route* une fois et une seule, et revenir à son point de départ ?

8.11.7

On considère un graphe G ayant $\mathbb{Z}/n$ comme ensemble de sommets, et pour arcs les couples d'entiers (i, succ(i)), avec succ(i) = (i + k) mod n.

En considérant les sommets de G sur un cercle, montrer que $k>n/2$ correspond à $n-k$, au sens de rotation près.

On pose pgcd(k,n) = p.

Montrer que si p = 1, G est un circuit eulérien, et que ce circuit correspond au tracé « à main levée » d'un polygone (étoilé si $k>1$) à n sommets reliés de k en k.

Montrer que si $1<p<n$, il n'y a plus de circuit eulérien, mais qu'il existe un polygone étoilé à n sommets formé de l'union de p polygones étoilés disjoints (ex : système à 15 sommets joints de 6 en 6).

8.12 PROBLÈMES

8.12.1

Partant du procédé matriciel pour déterminer l'existence de chemins, en déduire un procédé similaire pour déterminer l'existence de circuits autres que les boucles.

8.12.2

Chercher le schéma de programme le plus économique pour calculer une matrice ${}_{13}C_{19}$.

8.12.3 Compilation (J. Beney, 1985)

On considère un programme Π, formé d'un ensemble de procédures p, écrit dans un langage les acceptant déclarées dans n'importe quel ordre. Le problème est de traduire tout Π dans un langage où l'on impose, sauf pour la récursivité directe, que toute procédure soit définie après les procédures qu'elle emploie.

Définir un graphe d'appel inter-procédures. Lui appliquer une forme assouplie de fonction ordinale. En déduire un principe d'ordonnancement du code objet des

procédures. Caractériser la structure des programmes que ce procédé ne permet pas de traduire.

Même problème, sachant que le langage-cible impose, sauf pour la récursivité directe, qu'en tout point d'un programme, il n'y ait jamais plus de k procédures déjà utilisées et non encore définies (k = 4 pour le dialecte Pascal considéré comme cible).

8.12.4 Initiation à la méthode du chemin critique (ordonnancement, VLSI...)

On considère un projet P, formé d'un ensemble de tâches t, définies par des listes (d, t_1, t_2, t_3, ... t_n), qui fixent pour une tâche t sa durée d et la liste t_1, t_2, t_3, ... t_n des tâches à terminer avant que t commence.

On introduit des événements, fin ou début d'une ou plusieurs tâches, et on construit un Graphe d'Ordonnancement à raison d'un sommet par événement et d'un arc par tâche, cet arc étant valué par la durée.

Abstraction faite des durées, donner un critère de validation globale des données s'appuyant sur la structure du Graphe.

Montrer que le Graphe accepte nécessairement une fonction ordinale. Indiquer comment le tracer en rapport avec le/un chemin de plus grande taille, notion autorisée par l'absence de circuit.

On s'intéresse au projet défini par la table ci-après.

tâche	durée	prérequis
a	10	aucun
b	9	aucun
c	15	a
d	10	a
e	9	b
f	17	b
g	9	d, e
h	10	d, e
i	9	e, g
j	10	h, f

Après avoir défini le début du projet, la fin du projet et 5 autres événements convenables,

- établir le graphe correspondant, à raison d'un sommet par événement et d'un arc par tâche,
- calculer sa fonction ordinale,
- le tracer à l'aide de cette dernière.

On associe à chaque sommet/événement une date au plus tôt, 0 pour le début de projet.

A chaque tâche on associera :

- une date de lancement au plus tôt, qui sera la date de l'événement dont elle est issue,
- une date de fin au plus tôt, égale à sa date de lancement plus sa durée.

Les dates au plus tôt de chaque événement seront calculées de proche en proche,

- soit comme la fin au plus tôt de l'unique tâche qui y arrive,
- soit comme max des fins au plus tôt des tâches qui y arrivent.

Quelle est la durée minimale du projet ?

Le projet doit être fini au temps 40 : comment définir pour chaque tâche sa ***date de début au plus tard*** ?

On appelle ***chemin critique*** tout chemin ayant pour durée la durée maximale, car tout ralentissement d'une de ses tâches ralentit le projet tout entier.

Définir le chemin critique du cas considéré.

Que peut-on dire des tâches qui n'appartiennent pas à ce chemin ?

Montrer qu'on peut visualiser toutes les caractéristiques temporelles du projet, en bâtissant un chronogramme autour du /d'un chemin critique pris comme axe horizontal.

On envisage de modifier les ressources attribuées au projet. A quelles tâches devrait-on allouer prioritairement des ressources pour l'accélérer ? Sur quelles tâches peut-on faire sans risques les premières économies ?

8.12.6 Codes de Gray

Si la position angulaire d'un organe tournant est codée par des capteurs selon la numération usuelle, il y a notamment risque d'oscillation entre des positions de code o1111 et de code 1oooo, ou bien 11111 et ooooo. Si les différents bits ne sont pas alors établis de façon rigoureusement synchrone, tout autre code peut surgir de façon fugace, pouvant entraîner de graves erreurs.

On utilise donc des codes s'affranchissant de ce risque, comme dans la suite {oo, o1, 11, 1o, oo...}, où chaque transition ne modifie qu'un bit.

Pour cela, on exploite la notion de « distance de Hamming » entre 2 codes binaires, nombre de bits par lesquels les 2 codes diffèrent.

On veut définir un code de Gray à 16 « moments ».

Pour cela, on considère un graphe complet à 16 sommets régulièrement répartis sur un cercle et numérotés 0 à 15, chaque arête portant la distance de Hamming entre les codes binaires de ses extrémités. (ex : $d(3,6) = d(0011_2, 0110_2) = 2$).

Définir un code G4 en identifiant un *cycle hamiltonien* à 16 sommets, de longueur minimale, commençant par {0000, 0001, 0011, 0010, 0110...}. Montrer que chaque code ne diffère de ses deux voisins que par un bit ; qu'en particulier, le premier et le dernier code ne diffèrent que par un bit.

8.12.7 Enumération de codes par défilement

On se propose d'énumérer toutes les combinaisons d'un code par décalage. Un code donné p comprenant k bits, son successeur q est formé des k – 1 bits de poids faible de p, suivi d'un nouveau bit.

Ainsi, un code de valeur p a pour successeur un code de valeur 2p ou 2p + 1, modulo 2^k.

Construire le graphe des successeurs possibles pour les codes 0 à 7 (resp. 0 à 15).

En adaptant la recherche en profondeur, montrer que ces codes sont sur un même circuit hamiltonien, permettant de les énumérer en une seule suite.

Ecrire la suite des codes sous la forme d'une table de nombre binaires, et observer la propagation des 1 en diagonale.

Pour un système à k chiffres en base B, montrer que l'énumération simple demande kB^k chiffres, tandis que l'énumération par défilement n'en demande que $k - 1 + B^k$.

Quelle serait la longueur d'une chaîne contenant tous les identificateurs à 6 caractères ?

8.12.8 Rythmes (Prather)

Le mot sanscrit *yam'at'ar'ajabh'anasalag'am* est destiné à la mémorisation des 8 rythmes ternaires. Pour l'expliquer :

- transcrire ce mot en une chaîne binaire, en assignant 1 aux syllabes accentuées, o aux autres ;
- interpréter cette chaîne comme l'énumération par défilement des combinaisons d'un code.

8.12.9 Graphe d'implication

Soit un graphe G = (X,U) qu'on interprète ainsi : chaque sommet représente une proposition, chaque arc représente une implication établie.

Que signifie dans un tel graphe un chemin ? un circuit ? un chemin élémentaire ?

Considérons ce graphe comme une donnée à compléter.

Etablir un parallèle entre :

a) « rendre G réflexif » et une tautologie connue,

b) « compléter un jeu d'implications par des syllogismes et sorites » et « compléter G pour le rendre transitif ».

8.12.10 Déplacements urbains

On considère une agglomération de 10 km · 10 km, comportant environ 10 000 intersections et carrefours, et environ autant de segments de rues.

On lui associe d'abord un graphe G sur la base d'un sommet par intersection ou carrefour, et d'un arc ou d'une paire d'arcs par segment de rue, selon qu'il est ou non en sens unique.

Evaluer dans ce contexte les performances et exigences des méthodes de recherche de chemin point à point que vous connaissez.

On introduit maintenant entre sommets une relation d'équivalence « être du même quartier », en dédoublant éventuellement certains d'entre eux, de telle façon que le graphe-quotient associé rende essentiellement compte des liaisons inter-quartiers (ponts, tunnels, boulevards, autoroutes urbaines...).

En notant Q(a) le quartier de a, on propose de transformer le problème « aller de x à y » en 3 problèmes :

- un problème central « aller de Q(x) à Q(y) » ;
- un problème initial « dans Q(x), aller de x à la bonne limite de Q(x) » ;
- un problème final « dans Q(y), aller de la limite de Q(y) atteinte à partir de Q(x), à la destination finale y ».

Etudier les conditions de validité de cette approche, et définir le graphe exploité par chacune des phases. Montrer qu'on peut en espérer une accélération notable.

8.12.11 Tournois

Un graphe T = <X, R> est dit (graphe de) ***tournoi*** si c'est un graphe tel qu'entre tout sommet x et tout sommet y différent de x, il existe soit un arc (x, y) soit un arc (y, x). Il représente les résultats d'un tournoi dans lequel chaque compétiteur x affronte une fois et une seule chacun des autres y, la rencontre donnant soit x bat y, soit y bat x.

Etablir que chaque (graphe de) tournoi contient au moins un chemin hamiltonien.

Montrer que chaque tournoi induit au moins un classement sans ex aequo, en montrant que chaque graphe de tournoi est compatible avec au moins un ordre total.

Montrer que l'organisation d'un tel tournoi a une complexité de l'ordre de $n^2/2$.

On s'intéresse à des *tournois simplifiés,* représentables par des graphes partiels de T, qui, contenant encore un chemin hamiltonien, mèneraient à un classement total à un moindre coût.

Montrer qu'en s'inspirant des *tris par insertion* on peut obtenir des tournois simplifiés avec des complexités de $n^2/4$, $n^2/8$ voire $n \cdot \log_2 n$.

8.13 THÈMES DE RÉFLEXION

8.13.0 1001 algorithmes sur les graphes

Afin de disposer de *1001 algorithmes sur les graphes*, réaliser une bibliothèque générique de traitement matriciel style Dioïde dans le langage objet de votre choix (conseil : étudier d'abord GONDRAN & MINOUX, chap. I à III ; chap. 14 de ce livre).

8.13.1 Poèmes eulériens (de Werra, EPFL)

On considère un graphe, portant un alexandrin sur chaque arc. Montrer que chaque circuit eulérien mène a un poème différent. Caractériser cette différence.

Quelles autres contraintes observer pour améliorer ces poèmes ?

Peut-on remplacer les alexandrins par un autre type de vers ?

8.13.2 Charlemagne, les extra-terrestres et vous

En supposant que toutes les informations généalogiques souhaitables soient disponibles, et en utilisant les notions d'entrance et sortance moyennes d'un sommet, montrer à l'aide d'arguments démographiques qu'il sera moins coûteux de décider si Charlemagne fait partie de vos ancêtres[7] que de décider si vous faites partie des descendants de Charlemagne.

En déduire également que, même en admettant une lourde consanguinité, la plupart de vos ancêtres contemporains de Charlemagne étaient des extra-terrestres.

8.13.3 Couleurs et dégradés

Dans un système RVB, toute couleur est repérée par un triplet de chiffres représentant respectivement l'intensité des composantes rouge, verte et bleue. Ainsi, 000 représente-t-il toujours le noir, le blanc étant représenté par 999 en base décimale (ou 777 en octal).

Montrer que ces codes permettent d'assimiler une couleur à un point dans un Cube des Couleurs.

Introduire une notion de distance entre couleurs.

Lier la notion de *dégradé* d'une couleur à une autre à une notion de *chemin* dans le cube des couleurs.

Définir un *chemin progressif unitaire* tel que la transition de (k,l,m) à (p,q,r) se fasse en $|p-k| + |q-l| + |r-m|$ étapes, avec évolution monotone de chaque composante.

Combien y-a-t-il de tels chemins entre deux points ?

Définir en détail les 6 chemins progressifs unitaires les plus simples.

En vue de réaliser un dégradé harmonieux avec un nombre réduit de couleurs intermédiaires, définir de façon précise une notion de chemin diagonal entre couleurs, et en évaluer la pertinence.

Représentation des surfaces gauches : éclairage « au soleil » ou « au flambeau », ombrage progressif et faux relief.

8.14 LECTURES

BERGE C., *Théorie des graphes et ses applications,* CUM n°2, Dunod, 2e ed., 1963.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien,* tome I, chap. VI, Coll. Programmation, Ed. Gauthier-Villars, 1973.

[7] Entre Charlemagne et vous, compter 35 à 40 générations ; entre Clovis et vous, 45 à 50.

GONDRAN & MINOUX, *Graphes et algorithmes*, chap. III, collection EDF n°37, Eyrolles, 1979-95...

KAUFMANN A., BOULAYE G., *Théorie des treillis en vue des applications*, chap. I, Masson, 1978.

KAUFMANN A., DESBAZEILLE G., *La méthode du chemin critique,* Dunod, 1969.

KUNTZMANN J., *Théorie des réseaux,* Dunod, 1972.

LAURIÈRE J.C., *Intelligence artificielle : tome 1 : Résolution de problèmes,* Eyrolles.

MACOWICZ M., *Approche générique du traitement des graphes,* thèse INSA Lyon, novembre 1997.

PAIR C., DERNIAME J.C., *Problèmes de cheminement dans les graphes,* Monographie Afcet, Dunod, 1972.

REINHARDT F. et SOEDER H., *Atlas des mathématiques,* pp. 250 et sq., La Pochothèque, 1997.

ROY B., *Algèbre moderne et théorie des graphes,* Dunod, 1970, tome 2.

coll., *Petite encyclopédie des mathématiques,* chap. XXXVI, Edition Pagoulatos, Didier distr.

CHAPITRE 9

FERMETURES TRANSITIVES

Ce chapitre traite de la globalisation de relations. Cette élaboration produit des connaissances globales à partir de connaissances locales, et facilite souvent l'interprétation de relations quelconques, constatées dans des expériences ou des enquêtes.

9.1 FERMETURE TRANSITIVE ET DESCENDANCE STRICTE

DÉFINITION 9.1 – Parmi les relations définies sur un ensemble X, une relation A est dite ***fermeture transitive*** d'une relation R, et notée $A = R^{\oplus}$ si chaque fois l'on a x A y, on a

- soit x Ry,
- soit un z tel que x A z d'une part, z A y d'autre part.

$\mathbf{A=R^{\oplus}} \equiv$ A ***fermeture transitive de*** $R \equiv A = \{(x\ y) \in X^2 \mid x\ R\ y \vee (x\ A^2\ y)\}$

EXEMPLE. Considérons un ensemble P de personnes muni de la relation *R* = *estPereOuMereDe*. On globalise cette information, en introduisant la relation *A* = *estAncêtreDe*

- si x est père ou mère de y, x sera dit ancêtre (à l'ordre 1) de y,
- si x est ancêtre (à l'ordre k) de y, et y ancêtre (à l'ordre m) de z, alors x est ancêtre (à l'ordre k+m) de z.

En termes d'***algèbre des relations***, on notera $A = R \cup (A \bullet A)$
ou, sous une forme équivalente moins touffue : $A = R \cup (R \bullet A) = R \bullet (I \cup A)$

PROPOSITION 9.1 – *$A=R^{\oplus}$ est transitive,* sans que R ait à l'être.

DÉFINITION 9.2 – $R^{\oplus}x$, image de x par $R^{\oplus}$, sera dit ***descendance*** *de x (au sens de R)* ; le vocable ***cône de descendance*** mettra l'accent sur l'absence de circuits.

De même, y sera dit appartenir à la descendance de x au sens de R si xAy (resp. $xR^{\oplus}y$).

NOTE. Si l'on pose R = ilExisteUneRouteDirecte, alors A = ilExisteUneRoute renseigne sur l'accessibilité (sect. 10.4) ou la connexité (chap. 11). Si le graphe de R est valué par les longueurs des liaisons directes, [A] mènera à une table des distances au mieux (voir aussi chap. 14, les dioïdes).

9.2 FERMETURE RÉFLEXO-TRANSITIVE ET DESCENDANCE LARGE

DÉFINITION 9.3 – Soit maintenant $R^* = R^{\oplus} \cup I$ la relation ***être identique à ou être ancêtre de,*** *au sens de R.*

Cette relation, réflexive et transitive, et sera dite ***fermeture réflexo-transitive*** de R .

Elle est solution d'équations aux relations telles que : $R^* = I \cup (R \bullet R^*)$.

Dans le style de la définition 9.1, on définirait une version réflexo-transitive de A par

- tout x est ancêtre de lui-même (à l'ordre 0) ;
- si x est pereOuMere de y, et y ancêtre (à l'ordre m) de z, alors x est ancêtre (à l'ordre m + 1) de z.

R^* et $R^{\oplus}$ ne se distinguent que si $R^{\oplus}$ n'est pas réflexive, i.e. s'il existe au moins un sommet du graphe associé n'appartenant à aucune boucle ni circuit.

R^* est à la fois la fermeture la plus riche en propriétés et la plus facile à calculer (cf. Méthodes Matricielles au chapitre précédent) ; c'est pourquoi elle est souvent préférée à $R^{\oplus}$.

La fermeture réflexo-transitive R^* attache à un élément x un ensemble R^*x dit ***zone d'influence*** ou ***descendance au sens large*** de x (x en faisant partie).

On pourra noter $C^*_x = \Gamma^*x \cap \Gamma^{*-1}x$ l'ensemble des sommets formé de x et des sommets qui sont sur un même circuit que lui : c'est la classe de x pour l'équivalence $\approx_R \equiv R^* \cap R^{*-1}$.

Sauf raisons très spécifiques, où la question de savoir si un sommet appartient ou non à un circuit est critique, R^* se calculera à partir de R, si elle est réflexive, ou $I \cup R$, si R ne l'est pas : nous avons vu au chapitre précédent que cela nécessitera de l'ordre de $\lceil \log_2 n \rceil \cdot n^3$ opérations.

9.3 INTERPRÉTATION DES RELATIONS QUELCONQUES

9.3.1 Préordre le plus fin majorant une relation R

Comme indiqué en section 5.6, la première étape en vue de l'interprétation d'une relation quelconque R sur un ensemble X consiste à chercher un préordre représentant R « au mieux ». Nous avons justement :

PROPOSITION 9.2 – R est plus fine que sa fermeture réflexo-transitive R^*, et R^* est le plus fin des pré-ordres majorant R au sens de la relation de finesse.

9.3.2 Equivalence et partition associées à une relation

PROPOSITION 9.3 – L'équivalence $\approx_R = R^* \cap (R^*)^{-1}$ attachée au pré-ordre R^* est l'équivalence la moins fine minorant R^*.

Pour l'obtenir

- on part de la matrice M de R^*,
- on obtient chaque terme de la nouvelle matrice E associée à l'équivalence par

$$E[i, j] = M[i, j] \wedge M^T[i, j] = M[i, j] \wedge M[j, i]$$

car la transposée de la matrice associée à la relation d'origine est la matrice associée à la relation inverse (prop. 2.5).

[E] représentant une équivalence, doit être une matrice *diagonale par blocs*, ce qu'on obtient de fait par une permutation judicieuse de lignes et de colonnes. Les blocs diagonaux fournissent alors la partition $X/\approx_R$ de l'ensemble de départ au sens de (l'équivalence $\approx_R$ dérivée de) R.

2 cas extrêmes peuvent apparaître.

L'*équivalence* à l'arrivée est *réduite à l'identité* si au départ chaque sommet est isolé.

L'*équivalence* à l'arrivée est *totale* si au départ le graphe est *fortement connexe* (sect. 10.2).

Sinon, l'équivalence « quelconque » obtenue structure l'ensemble de départ en classes d'objets comparables, dont chacune deviendra sommet du graphe-quotient $X/\approx_R$ (§ 8.2.1).

9.3.3 Ordre associé à une relation R

Nous avons vu au paragraphe 5.6.4 qu'un préordre R* induisait une relation d'ordre $\angle_R$ sur l'ensemble-quotient $X/\approx_R$ dérivé de R via l'équivalence $\approx_R$ attachée à R*.

9.3.4 Niveaux de lecture d'une relation quelconque

La séquence relation quelconque / fermeture réflexo-transitive / équivalence / ensemble-quotient / ordre ramène au niveau macroscopique pratiquement toute relation à un ordre partiel entre groupe d'objets comparables.

Un examen ultérieur permet de caractériser plus finement R par les déficiences locales : intermédiaires obligés dont le rôle disparaît quand la transitivité est ajoutée, points d'entrée ou de sortie d'une classe d'équivalence, circuits élémentaires de tailles diverses dans une même classe.

9.3.5 Exemple : un sondage

On a demandé à un échantillon de N personnes de dire leurs préférences relatives entre 7 poudres à laver (ou vedettes de la chanson).

Le graphe G ci-dessous rend compte des *préférences déclarées plus de n fois* (fig. 9.1).

La relation correspondante n'a pas a priori de propriété « sympathique » la rattachant directement à une catégorie claire.

La réflexivité est arbitraire, la relation est de plus non transitive et non symétrique (n'étant asymétrique que si n > N/2).

Cependant, le procédé s'applique.

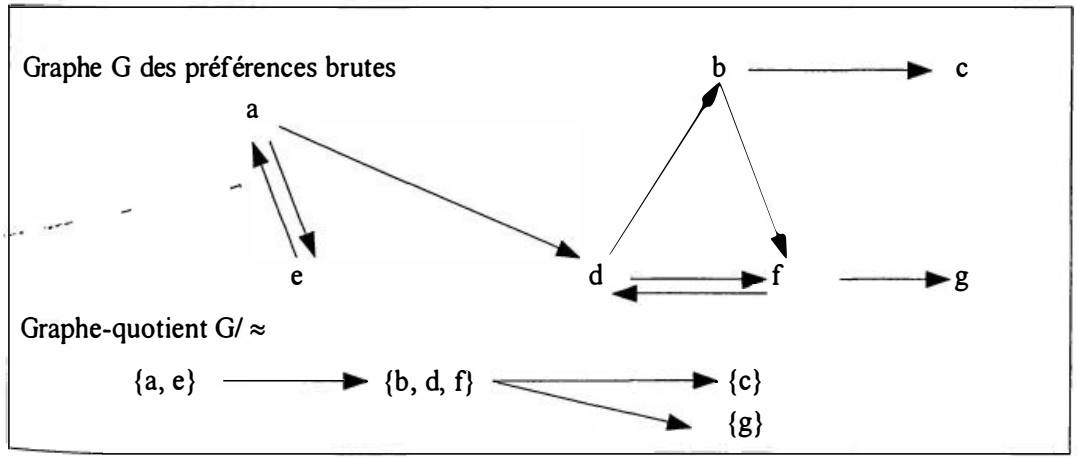


Fig. 9.1 Graphe de préférence et son graphe-quotient.

Tableau 9.1 Relation, fermeture transitive et équivalence associées.

R	a	b	c	d	e	f	g
a	o	o	o	1	1	o	o
b	o	o	1	o	o	1	o
c	o	o	o	o	o	o	o
d	o	1	o	o	o	1	o
e	1	o	o	o	o	o	o
f	o	o	o	1	o	o	1
g	o	o	o	o	o	o	o

⇒

R*	a	b	c	d	e	f	g
a	1	1	1	1	1	1	1
b	o	1	1	1	o	1	1
c	o	o	1	o	o	o	o
d	o	1	1	1	o	1	1
e	1	1	1	1	1	1	1
f	o	1	1	1	o	1	1
g	o	o	o	o	o	o	1

construction de la plus grande
sous-matrice symétrique
puis
dégagement de blocs diagonaux
par permutation de lignes et colonnes

⇒

≈	a	e	b	d	f	c	g
a	1	1	o	o	o	o	o
e	1	1	o	o	o	o	o
b	o	o	1	1	1	o	o
d	o	o	1	1	1	o	o
f	o	o	1	1	1	o	o
c	o	o	o	o	o	1	o
g	o	o	o	o	o	o	1

EN CONCLUSION

- sont préférés : (a, e) d'abord, puis (b, d, f), enfin c d'une part, g d'autre part (c et g restant non comparables) ;
- le sondage est interprété avec un seuil n assez bas puisqu'il y a des cas de symétrie : au moins n personnes préfèrent a à e, et e à a ; de même pour d, f ; si les ex aequo dans les classements sont interdits, alors $n < N/2$;
- en retraitant les données d'origine avec un seuil un peu plus élevé, y a-t-il d'abord un clivage (b, d)(f) ou (a)(e) ?

9.4 ACCESSIBILITÉ DANS LES SYSTÈMES À ÉTATS

9.4.1 Les systèmes à états

Les systèmes dynamiques peuvent souvent être modélisés par un graphe

- dont les sommets représentent des *états* plus ou moins stables, plus ou moins fugaces ;
- dont les arcs représentent les *transitions* possibles (marquables par une éventuelle cause interne ou externe : commande, entrée).

De tels graphes permettent de dégager les grandes tendances du comportement de tels systèmes.

En regroupant les états en classes présentant une certaine stabilité et une certaine similitude de rôles, on facilite la compréhension en donnant la possibilité de distinguer deux niveaux : *fonctionnement global* et *comportement fin*.

EXEMPLE. Le système ci-après (fig. 9.2), partant de 0, parcourt les états de la classe A pendant la phase initiale. Ensuite,

- soit il passe dans la classe B, pour finir soit dans la classe C soit dans la classe E,
- soit il passe dans le classe D pour finir dans la classe E.

Les états de la classe A forment un cycle et correspondent par exemple à une mise en route, dont la fin est soumise à conditions.

Une transition se produit

- vers les états B si (a) ... (transition u → v), ou si (b) ... (transition z → w),
- vers les états C si ... (transition z → t).

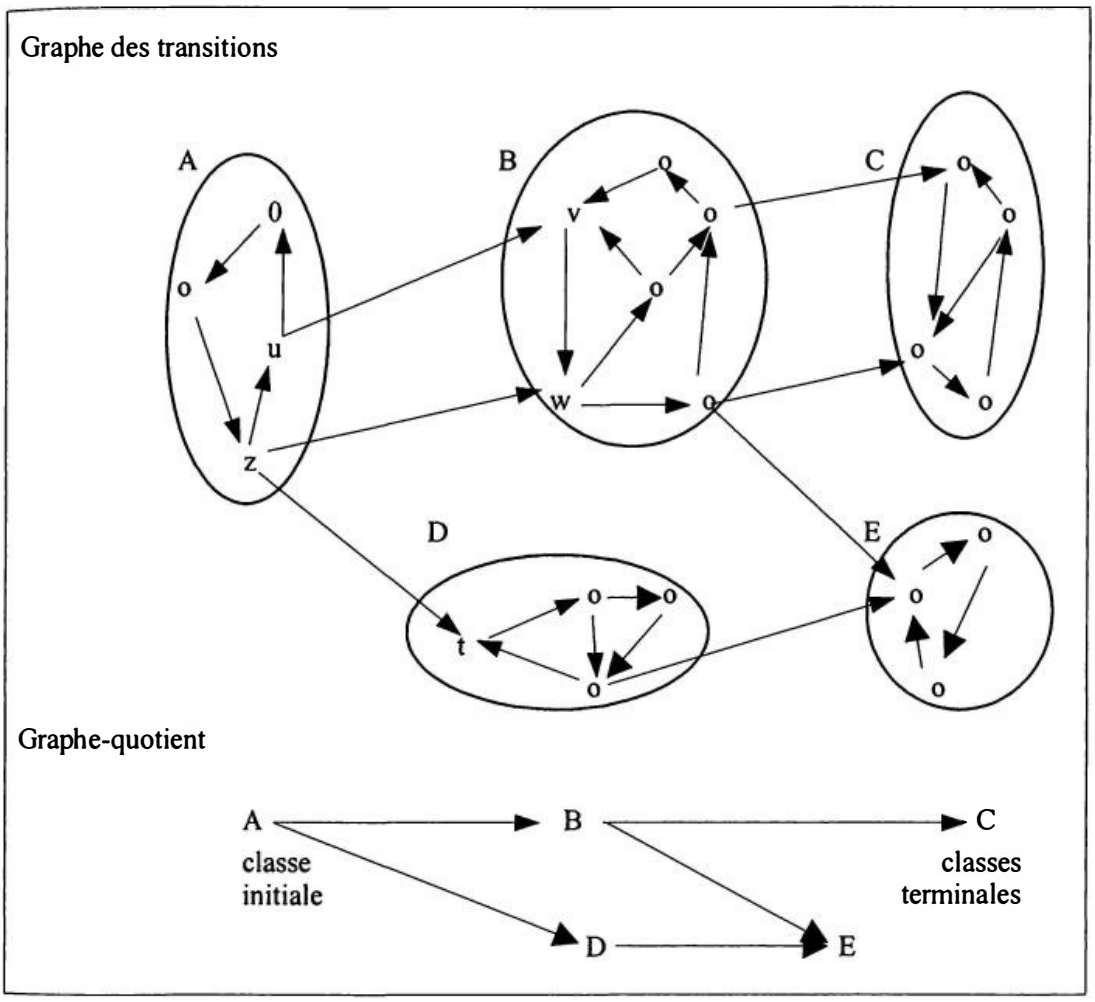


Fig. 9.2 Graphe des transitions et graphe-quotient d'un système à états.

9.4.2 Commandabilité des automatismes

Les automatismes à séquence, genre commande d'ascenseur, peuvent souvent être modélisés par un graphe

- dont les sommets représentent des états,
- dont les arcs représentant les transitions possibles, sont porteurs des commandes ou entrées déclenchant la transition.

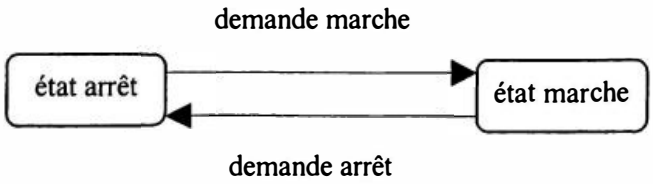


Fig. 9.3 Automate minimal.

En particulier pour les redémarrages après incident, on s'intéresse souvent à l'*accessibilité d'un état y à partir d'un autre x* : pour cela, il faut qu'il y ait un chemin dans le graphe de x à y, et la suite des entrées lues le long de ce chemin donne la liste des commandes ou entrées qui réaliseront la transition ; la multiplicité de tels chemins permettra de choisir le plus facile à mettre en œuvre.

Pour éviter de tels problèmes, il arrive qu'un circuit général rende de fait tout état accessible de tout autre, quitte à passer par une séquence d'arrêt et de redémarrage.

Après un incident, il arrive que l'état exact du système ne soit pas connu. L'observation des sorties doit alors permettre de définir l'ensemble des états dans lesquels il peut se trouver, et de là le jeu de commandes le ramenant à un état standard, en évitant tout nouvel incident. La poursuite de l'observation des sorties et des réactions accélère la levée des ambiguïtés, clarifiant et sécurisant la remise en état.

La structure du graphe peut cependant indiquer qu'il n'existe pas un tel chemin, si l'état x est dans une classe qui ne précède pas celle de y dans le graphe-quotient : x est alors exclu du cône des prédécesseurs de y, qui n'est plus accessible de x.

Tableau 9.2 Circuits et accessibilité.

commande de à	arrêt	initial	marche réduite	marche normale	incident	purge
arrêt	∅	M				
initial		∅	p4			
marche réduite			–, ∅	+	a, b	A
marche normale			–	+, ∅	b, c, z	
incident			b99		...	w
purge	p99					...

On peut supposer M, A, +, – commandes externes, les autres « commandes » pouvant être des comptes rendus internes (ex : b99 : reprise à marche réduite quand le système a su résoudre un incident de type b...).

Tout état est accessible, des circuits particuliers se greffant sur un circuit marche/arrêt général.

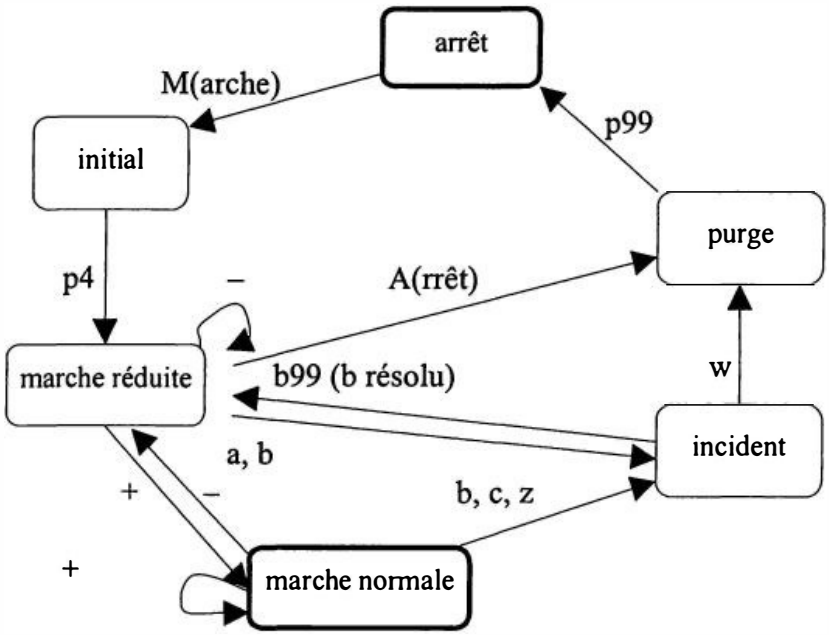


Fig. 9.4 Graphe de transition d'un automate et accessibilité.

9.4.3 Processus de Markoff

Dès 1912, Markoff formulait l'hypothèse que, dans certains systèmes dynamiques, il fallait voir la probabilité d'un évènement comme fonction de l'évènement antérieur plutôt qu'en soi, et vérifiait cette idée par le comptage de lettres et de digrammes dans *Eugène Oniguine* de Pouchkine.

En français, on constate que la lettre u constitue 6% des lettres des textes, mais que sa probabilité d'occurrence derrière un q est très élevée.

Considérons un processus aléatoire à tirages indépendants : si un signe s est émis avec une probabilité p indépendante du contexte, la longueur moyenne d'une suite de signes s sera $1/(1-p)$.

Si, face à une émission de signes s ayant une probabilité p, on constate l'apparition de séquences longues, une approche markovienne est plus indiquée.

On représente usuellement un processus de Markoff par la matrice carrée des p_{ij}, probabilité de transition à l'état j lorsqu'on est dans l'état i. Ces matrices ont donc (a) tous leurs éléments valant de 0 à 1 inclus, (b) toutes leurs lignes de somme 1.

On leur associe un graphe, à raison d'un sommet par état, et d'un arc (i, j) par probabilité $p_{ij} > 0$.

Sur le long terme, l'évolution du processus ne dépend que de la seule présence/absence d'arc, qui régit l'accessibilité d'un état, lequel devient un piège si $p_{ii} = 1$. La partition issue de la fermeture transitive fournit les grandes classes d'états, stables si elles sont terminales, instables si elles ne comportent qu'un sommet sans boucle à l'origine, métastables dans tous les autres cas. Le graphe-quotient donne alors les tendances du processus :

- fonctionnement permanent entre tous les états, s'il existe un circuit hamiltonien,
- ou fonctionnement se restreignant à une composante fortement connexe, la classe terminale.

Les *valeurs numériques* des probabilités *fixent la vitesse d'évolution du processus, la durée moyenne* de maintien dans un état donné...

EXEMPLE. Pour une personne dont on modélise l'humeur, normale ou irritée, un modèle markovien rendra compte

- de la fréquence relative des états « normal » et « irrité »,
- de la durée moyenne de l'irritation (rendue indépendante de la fréquence),
- par contrecoup, de la susceptibilité de la personne.

9.4.4 Structure d'influence et physique qualitative

On sait bien mettre en équations la trajectoire d'un objet, mais personne n'utilise ces équations quand il s'agit d'envoyer une boule de papier dans une balle à papier ; or un tel jet peut cependant être réussi ou, par essais successifs, s'améliorer rapidement.

L'idée est venue à divers auteurs qu'il devait y avoir une sorte d'appréhension intuitive de la situation, cependant suffisante. On a proposé depuis un modèle à base de graphes, à raison d'un sommet par grandeur perçue, et d'un arc de x vers y marqué +, absent, ou marqué –, selon le signe et la valeur de $\partial y/\partial x$, l'absence correspondant à une influence directe négligeable.

L'existence d'un chemin de x vers y témoigne d'une influence moins directe ; on la qualifie de positive ou négative selon que le chemin comporte un nombre pair (voire nul) ou impair d'arc(s) négatif(s).

Les deux difficultés soulevées par cette approche concernent les circuits, et les agrégations de chemins de signes opposés.

Il est parfois difficile d'interpréter qualitativement les circuits, dès lors que la théorie de la rétroaction montre que la distance entre régime stable et régime instable est souvent très faible et demande une appréciation numérique.

Il est également délicat de savoir ce qui se passe si de x à y il y a plusieurs chemins de signes différents : le(s)quel(s) l'emporte(nt) globalement?

Cependant, on peut déjà se faire une idée des interdépendances du système, et de ses parties critiques.

De tels graphes sont aussi utilisables en économie ou en sociologie avec une interprétation semblable.

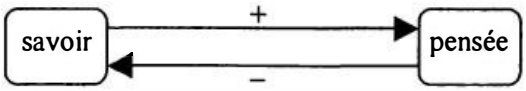


Fig. 9.5 Graphe pour « plus je sais, plus je pense ; plus je pense, moins je sais ».

9.5 EXERCICE

9.5.0

Soit R une relation transitive et irréflexive, à laquelle on associe la relation squelette $S = R - R^2$.

Montrer que $R = S^{\oplus}$.

9.6 PROBLÈMES

9.6.0 (d'après Huxtable, 1961)

On considère un programme Π utilisant un ensemble P de procédures p, et on considère le graphe G = (P, U) ainsi défini : il existe un arc (p1, p2) dans G ssi la procédure p1 appelle la procédure p2 dans le programme Π.

Revoir la signification en termes de procédures et de programme des chemins, des boucles et des circuits de G.

Soit à compiler Π. On suppose qu'on dispose de deux politiques de traduction des procédures, dont une très efficace qui ne s'applique qu'aux procédures non récursives. Qu'apporte l'exploitation de la fermeture transitive de G ?

Soit à fragmenter Π en modules. Qu'apporte la fermeture réflexo-transitive de G, l'équivalence et le graphe quotient dérivé ? Montrer que ce dernier graphe favorise le groupement des modules en couches.

9.6.1

A raison d'un sommet par quantité et d'un arc par influence, exprimer par un graphe à arcs signés, style physique qualitative, les observations suivantes :

1. Quand les prix grimpent, les salaires grimpent.
2. Quand les salaires grimpent, les prix grimpent.
3. Quand les prix grimpent, les volumes baissent.
4. Quand le volume croît, l'embauche croît.
5. Quand l'embauche croît les salaires croissent.

Mettre en lumière un circuit négatif et un circuit positif.

Selon les vitesses de parcours et les effets d'amortissement ou d'amplification d'une fluctuation, montrer que ces circuits provoquent soit une régulation, soit une instabilité, cyclique ou exponentielle.

Attacher à chacun de ces arcs des conditions de validité.

Vérifier qu'elles sont de différents ordres, expliquant des effets variables de cette structure selon les scénarios.

9.6.2 Sondage

La société Vox Populi est chargée d'enquêter sur les préférences d'utilisateurs pour une liste de produits. Dans un échantillon E de 50 utilisateurs, chaque personne interrogée donne un ordre de préférence pour 5 produits au plus. On dresse ensuite une matrice des préférences P, *chaque p_{ij} indiquant le nombre exact de sondés ayant classé i avant j* (tableau P ci-après).

On introduit une famille de relations P_k sur la base :

$x\ P_k\ y \equiv$ k personnes au moins préfèrent x à y.

Pour un seuil k décroissant de 50 à 0, définir :

- les relations P_k distinctes à retenir
- les fermetures transitives F_k associées,
- les interprétations correspondantes des P_k.

P	a	b	c	d	e	f
a	0	15	14	9	18	20
b	7	0	6	13	9	6
c	8	9	0	11	15	11
d	7	10	8	0	9	5
e	5	15	11	9	0	9
f	3	17	8	3	15	0

On peut étendre la distance de Hamming entre vecteurs booléens à une distance entre relations. Pour cela, on prend comme distance entre relations R et S sur un même ensemble, le nombre $|R\Delta S|$ de couples satisfaisant l'une mais pas l'autre. En ce sens, peut-on dire qu'une fermeture F_k est « peu distante », « bonne approximation » des P_k, P_{k-1}, P_{k-2} … ? Si $k > N/2$?

9.6.3 Graphe d'implication

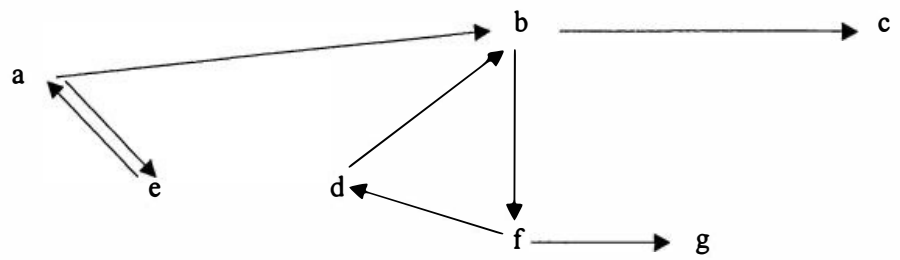
On considère un graphe G = (X,U) avec la convention : chaque sommet représente une proposition, chaque arc représente une implication établie.

Que signifient les fermetures transitives et réflexo-transitives de G :

- en tant que processus ?
- en tant que résultats ?

Que signifient les classes initiales, finales ou intermédiaires issues de la fermeture réflexo-transitive?

Graphe G d'implication



On considère le graphe ci-dessus. Montrer qu'il suffit d'établir 2 implications (parmi 8) pour prouver que toutes ses propositions sont équivalentes (à un sorite près).

9.6.4

On considère des images noir et blanc ayant une proportion x = 85% de pixels blancs et une proportion y = 15% de pixels noirs.

Montrer que cela devrait entraîner des séquences blanches de 6,7 pixels en moyenne.

On observe en pratique des séquences blanches d'une longueur moyenne de 20 pixels. On tente alors de décrire le phénomène par un processus de Markoff de matrice caractéristique donnée ci-contre.

Tracer le graphe valué complet associé.

$$\begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$$

En régime stationnaire, montrer que l'on a :

$$(x\ y)\begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix} = (x\ y)$$

d'où, finalement,

$$x = \frac{p}{p+q}; \qquad y = \frac{p}{p+q}; \qquad x/y = p/q$$

Montrer que les longueurs moyennes des séquences sont respectivement 1/p et 1/q.

En déduire les probabilités p, q associées à l'image décrite, et la longueur moyenne d'une séquence de pixels noirs.

On désire transmettre des images de 1 000 000 pixels. On souhaite coder les séquences à l'aide d'un ou plusieurs codes successifs formés d'un bit de couleur, suivi d'un champ de n bits ($n \geq 2$) contenant un nombre de 0 à $2^n - 1$ représentant une longueur de 1 à 2^n.

En fonction de n, combien demandent-ils de bits par pixel en moyenne ? Quel n donne la meilleure compression ?

9.7 LECTURES

BERGE C., *Théorie des graphes et ses applications,* Dunod, 1958-63.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien,* tome I, chap. VI, Coll. Programmation, Ed. Gauthier-Villars, 1973.

GONDRAN & MINOUX, *Graphes et algorithmes,* chap. III, collection EDF n°37, Eyrolles, 1979-85...

KAUFMANN A., PICHAT E., *Méthodes mathématiques non numériques,* vol. 2 Algorithmes de recherche de chemins et problèmes associés, Masson, 1977.

KUNTZMANN J., *Théorie des réseaux*, Dunod, 1972.

PAIR C., DERNIAME J.C., *Problèmes de cheminement dans les graphes,* Monographie Afcet, Dunod, 1972.

ROY R., *Algèbre moderne et théorie des graphes*, tome 2, Dunod, 1970.

CHAPITRE 10

ARBRES & CONNEXITÉ

L'*arbre* est un schéma fréquemment rencontré dans les structures de décision (questionnaires, hiérarchies, classifications) et de dénombrement.

Antérieure à la Renaissance, la notion de *connexité* exprime les liaisons et interdépendances, et rend compte de l'accessibilité d'un élément à un autre, considérée comme d'autant plus robuste qu'elle tolère des suppressions locales d'éléments ou de liaisons dans l'environnement des éléments concernés.

En ce sens, l'arbre est un graphe connexe, mais étant la moins redondante de ces structures, elle est la moins robuste.

10.1 CHAÎNES & CYCLES

10.1.1 Chaînes & chemins

DÉFINITION 10.1 – Deux sommets x et y d'un graphe G sont sur une même ***chaîne*** :

- s'ils sont adjacents,
- ou si z étant un autre sommet de G, il existe une chaîne de x à z, et une chaîne de z à y.

COROLLAIRE 10.1 – Dans un *graphe symétrique* :

- les notions de chaîne et de chemin sont confondues,
- la relation *être identique ou sur une même chaîne*, relation d'équivalence entre sommets du graphe, se confond avec *être identique ou sur un même circuit.*

PROPOSITION 10.2 – Dans un *graphe orienté* :

- La relation *être identique ou sur une même chaîne* est une relation d'équivalence entre sommets du graphe, moins fine que *être identique ou sur un même circuit.*
- Tout chemin suppose une chaîne, obtenue en négligeant l'orientation des arcs.
- Une chaîne ne peut devenir chemin que si chaque arête peut supporter une orientation commune.
- Ce graphe orienté possèdera une chaîne chaque fois que son symétrisé en possèdera une.

Fig. 10.1 Chaîne sans chemin.

EXEMPLE. Dans un ensemble de personnes structuré par la relation *aPourEnfant*,

- il y a *chemin* entre toute personne et chacun de ses descendants ;

- il y a *chaîne* dans ce cas, puis entre chaque personne et ses ancêtres, puis entre toutes les personnes qui partagent un ancêtre ou un descendant, et de proche en proche pour tous les couples (x, y) pour lesquels on peut passer de l'un à l'autre par des intermédiaires partageant un ancêtre ou un descendant avec le précédent et le suivant.

10.1.2 Cycles & circuits

DÉFINITION 10.2 – Un ***cycle*** est une chaîne ($x = x_0, x_1, x_2, \ldots x_n$) dont les extrémités coïncident, i.e. telle que $x_n = x$.

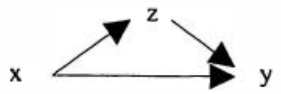


Fig. 10.2 Cycle sans circuit.

Les notions de cycle et de circuit sont confondues dans un graphe symétrique.

PROPOSITION 10.3 – Un graphe orienté possède un cycle *ssi* son symétrisé en possède un.

10.2 GRAPHE (FORTEMENT) CONNEXE

DÉFINITION 10.3 – Un graphe G = <X, U> est dit

- ***graphe fortement connexe*** si pour tous les sommets x, y de X, x = y ou il existe un *chemin* de x à y (ex : G_1, fig. 7.4) ;
- ***graphe connexe*** si pour tous les sommets x, y de X, x = y ou il existe une *chaîne* de x à y (ex : G_1, fig. 7.4 et G_2, fig. 8.3).

Ces notions coïncident pour les graphes symétriques. Sinon, la forte connexité *implique* la connexité.

10.3 COMPOSANTE CONNEXE D'UN GRAPHE

DÉFINITION 10.4 – On appelle ***composante connexe*** C d'un graphe G, tout sous-graphe connexe de G, maximal pour cette propriété (i.e. qui la perd si on lui ajoute un sommet).

Les *composantes connexes* sont les *classes de connexité,* classes d'équivalence de la relation *être identique ou sur une même chaîne.*

NOTES

- Dans un graphe G, la composante connexe C_x à laquelle appartient un sommet x s'obtient en recherchant tous les descendants de x
  - dans G, s'il est symétrique,
  - dans son symétrisé, si G est orienté.

  La recherche peut s'effectuer en exploitant la matrice relative à la fermeture transitive de la relation symétrisée, ou par une méthode locale.

- La *matrice d'adjacence d'un graphe* quelconque est *diagonale par blocs*, si ses lignes et colonnes ont été permutées pour regrouper ses sommets par composante connexe.

DÉFINITION 10.5 – On appelle ***nombre de connexité*** d'un graphe le nombre de ses composantes connexes.

NOTE. L'ensemble des composantes connexes se trouve en partant des sommets, au départ non-numérotés. Tant qu'il reste un sommet non numéroté : on lui donne un *nouveau* numéro, puis on assigne le *même* numéro à tous les sommets de sa composante connexe.

Ce processus se termine dès lors que le graphe est fini.

PROPOSITIONS

10.4 – Un ***graphe connexe*** est formé d'une seule composante connexe : son nombre de connexité est 1.

10.5 – L'existence d'un chemin de x à y suppose x et y dans la même composante connexe (par l'absurde).

10.4 COMPOSANTES FORTEMENT CONNEXES

Dans les graphes orientés, considérons les *sous-ensembles fortement connexes maximaux* : ce sont les ***composantes fortement connexes,*** classes d'équivalence fournies par la fermeture transitive, internes ou identiques aux différentes composantes connexes.

EXEMPLES

- Si un graphe représente une structure de communication, la communication est bilatérale entre x et y dès lors qu'ils font partie d'une même composante fortement connexe.
- Dans un réseau de communication en construction, on peut ainsi distinguer ce qui est fait (le graphe de communication) de ce qui est opérationnel (les composantes fortement connexes, classes d'équivalence pour « peut dialoguer avec ».)
- Si une municipalité veut introduire ou modifier un système de rues à sens unique, elle doit (faire) vérifier que l'ensemble des rues ainsi orientées, totalement ou partiellement, constitue toujours un graphe fortement connexe. Sinon, certains points de la ville deviendraient inaccessibles, d'autres se transformeraient en pièges...

Dans les graphes symétriques, il y a confusion entre composantes connexes et composantes fortement connexes. Sinon, les composantes fortement connexes partitionnent les composantes connexes.

10.5 CENTRES, RAYON, DIAMÈTRE D'UN GRAPHE

10.5.1 Ecart, écartement

DÉFINITION 10.6 – Soit $G = \langle X, U, L \rangle$ un graphe dont les arcs sont valués par une longueur. On appelle $d(x, y)$ l'***écart*** de x à y, ou *longueur du plus court chemin allant de x à y,* soumis aux conventions suivantes :

- pour tout $x \in X$, $d(x, x) = 0$,
- si $y \notin \Gamma^*(x)$, alors $d(x, y) = \infty$.

PROPOSITION 10.6 – (BERGE) d(x,y) vérifie :

- $d(x, x) = 0$
- $d(x, z) \leq d(x, y) + d(y, z)$
- et, si le graphe est symétrique, $d(x, y) = d(y, x)$

DÉFINITION 10.7 – On appelle ***écartement*** d'un sommet x : $e(x) \equiv \max_{y \in X} (d(x,y))$

REMARQUE. Un graphe qui n'est pas fortement connexe possède au moins un sommet d'écartement infini.

10.5.2 Centre(s) et rayon

DÉFINITION 10.8 – On appelle

- ***centre*** d'un graphe tout sommet d'écartement minimal,
- ***rayon*** d'un graphe son écartement minimal ρ.

NOTE. L'équivalent géométrique, pour une zone (resp. volume) connexe et convexe munie d'une distance euclidienne est donné par le centre et le rayon d'un cercle (resp. d'une sphère) *circonscrit* à la zone (resp. au volume) *et de rayon minimal*.

10.5.3 Diamètre

DÉFINITION 10.9 – Soit G un graphe fini, fortement connexe. On appelle ***diamètre*** δ de G son *écartement maximal* :

$$\delta = \max_{x \in X}(e(x)) = \max_{x, y \in X}(d(x, y))$$

PROPOSITIONS

10.7 – Un graphe non fortement connexe ou ayant une boucle aurait un diamètre infini.

10.8 – (BERGE) Dans un graphe fortement connexe sans boucle à n sommets et m arcs on a

- $n \leq m$
- $m \leq n \cdot (n - 1)$
- $\delta \leq (n - 1)$ (si on raisonne en taille)

10.9 – Dans un graphe symétrique $\delta \leq 2\rho$

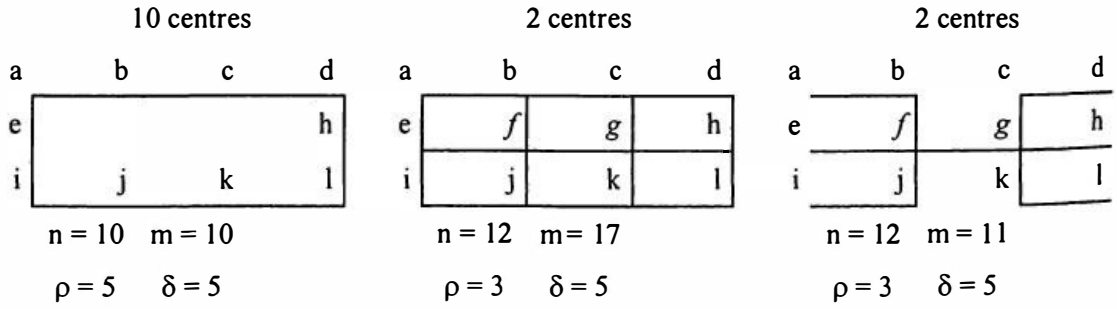


Fig. 10.3 Centres, rayons, diamètres.

10.5.4 Application aux réseaux

Supposons que G représente un réseau de communication.

- Le nombre n de sommets indique le *nombre de sites* desservis.
- Le nombre m d'arcs dénote le *nombre de liaisons directes* (coût).
- Le *rayon* ρ qualifie l'*éloignement des centres.*
- Le *diamètre* δ caractérise les *liaisons les plus longues.*

10.6 ARBRES, FORÊTS, HIÉRARCHIES, ARBORESCENCES

DÉFINITIONS

10.10 – Un ***arbre*** est un graphe symétrique connexe sans cycle.

Une ***forêt*** est un graphe symétrique dont chaque composante connexe est un arbre.

10.11 – Une ***hiérarchie*** est un graphe connexe orienté *sans circuits.*

Une ***arborescence*** est un graphe orienté, connexe et *sans cycle* possédant :

- un et un seul sommet initial, dit ***racine***,
- des sommets terminaux ou ***feuilles***,
- d'autres descendants, ayant un seul prédécesseur et au moins un successeur, dits ***nœuds.***

EXEMPLES

- Soit un ***questionnaire*** à N issues disjointes. Il est *arborescent* s'il existe un seul chemin(ement) de la question initiale à chacune des issues. Il est seulement *hiérarchique* s'il existe plusieurs entrées, plusieurs issues, pas de circuits mais des sous-chemins communs.
- Le système de fichiers d'unix est un graphe orienté connexe. La présence de circuits (liens . et ..) fait qu'il ne s'agit pas d'une arborescence (Y. Martinez).
- Une arborescence, plus souvent une hiérarchie, peut constituer un modèle utile, sur la base d'un sommet par situation et d'un arc par action pour
  - les ordonnancements,
  - les jeux,
  - les conflits,
  - les négociations (Rand Corp., affaire de Cuba, 1962),

  l'absence de circuit signifiant qu'on ne revient jamais exactement à une situation antérieure.
- En vue de résoudre un problème, on est souvent admis à le décomposer hiérarchiquement en sous-problèmes mieux connus, un circuit n'apparaissant que si un problème est défini récursivement (i.e. par rapport à une autre version de lui-même).

NOTE. Pour une transition naturelle de la notion d'arbre à celle d'arborescence, cf. Jacques VÉLU (1994), chapitre 20.

PROPOSITION 10.10 – Un graphe orienté sans circuit <X, Γ> possède deux ***fonctions ordinales*** g et h définies par :

$$\forall\, x \in X \;\; g(x) \equiv \text{si } \Gamma(x) = \varnothing \text{ alors } 0 \text{ sinon } 1 + \max_{y \in \Gamma(x)}(g(y))$$

$$\forall\, x \in X \;\; h(x) \equiv \text{si } \Gamma^{-1}(x) = \varnothing \text{ alors } 0 \text{ sinon } 1 + \max_{y \in \Gamma^{-1}(x)}(h(y))$$

NOTES

- L'absence de circuit garantit la convergence de la définition.
- Ces fonctions ordinales facilitent un *tracé clarifié* des graphes sans circuit. Considérons le graphe G_4 de la figure 10.4. En calculant g() pour chacun de ses sommets, on peut en proposer la représentation 10.5, qui regroupe sur un même niveau n les sommets x tels que g(x) = n, et met en évidence la hiérarchie correspondante (fig. 10.6 fait de même avec la fonction h).

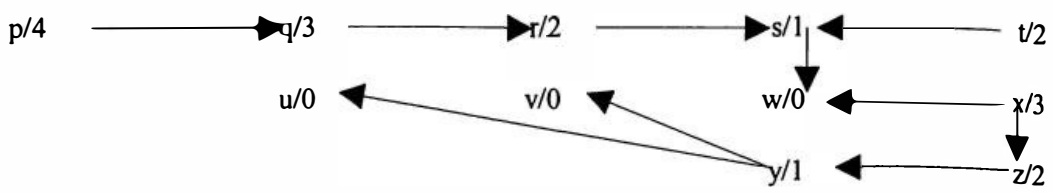


Fig. 10.4 Graphe sans circuit G_4 et sa fonction ordinale g.

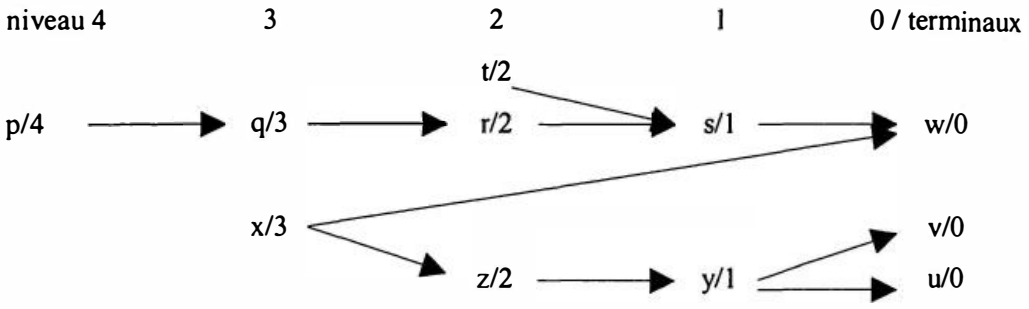


Fig. 10.5 G_4 redessiné par niveaux de sa fonction ordinale g.

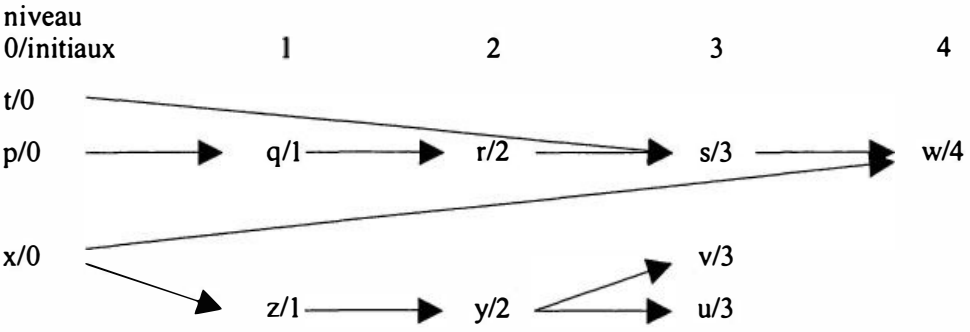


Fig. 10.6 G_4 redessiné par niveaux de sa fonction ordinale h.

10.7 ARBRE COUVRANT

DÉFINITION 10.12 – Soit un graphe symétrique connexe G = <X, U>. On appelle ***arbre couvrant*** de G tout arbre A = <X, T>, tel que :

- $T \subseteq U$,
- $\forall x \in X\ \exists y \in X\ (x, y) \in T$ (tout sommet de X est couvert par une arête de T).

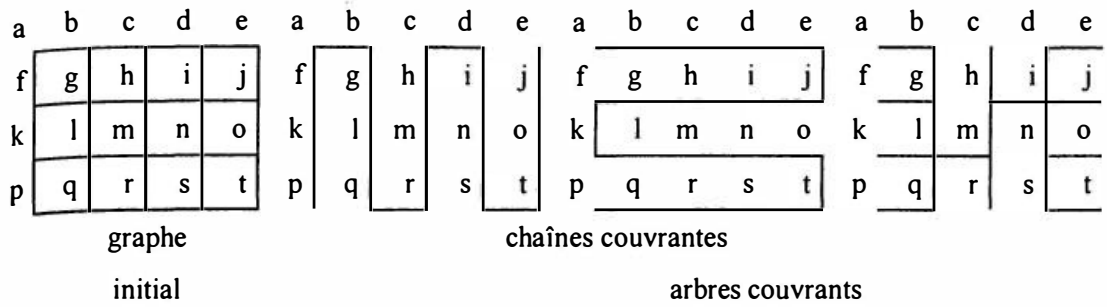


Fig. 10.7 Arbres couvrants.

10.8 ARBRE COUVRANT MINIMAL

Soit un ensemble de points à relier entre eux au moindre coût.

On se donne le graphe G, présumé connexe, des liaisons directes possibles, présumées symétriques et de coût connu. On veut en extraire un graphe partiel connexe R dont l'ensemble des arêtes soit de coût minimal.

PROPOSITION 10.11 – Soit un graphe symétrique connexe G, dont chaque arête est de de coût strictement positif. R, graphe partiel de G, connexe et de coût minimal, est un arbre, dit ***arbre couvrant minimal.***

PREUVE. Supposons que R, de coût minimal, possède un cycle. On pourrait enlever au moins une arête du cycle sans déconnecter R, et le nouveau graphe obtenu R^- serait un graphe partiel (de R donc de G) couvrant (les sommets de R donc de G), et de coût strictement inférieur à celui de R, contredisant l'hypothèse de coût minimal. R est donc sans cycle : c'est un arbre couvrant de coût minimal.

Pour bâtir un tel *arbre couvrant (de coût) minimal*, Kruskal a proposé le procédé suivant :

Recherche d'un *arbre couvrant (de coût) minimal* d'un graphe G = <X, U, μ : U→V> _

Former la liste L des arêtes de U triées par coût croissant, et la liste R, initialement vide, des arêtes du réseau cherché.

Définir l'ensemble Y initialement vide des points couverts, et l'ensemble Z = X – Y des points restant à couvrir.

Transférer de L dans R la première arête u de L, (i .e. la moins chère).

Transférer de Z à Y les deux sommets de u.

Tant qu'il reste un sommet dans Z,

- retirer de L et mettre dans R l'arête u_k la moins chère reliant un sommet de Y à un sommet de Z,
- transférer de Z à Y le sommet de u_k appartenant à Z

fin Recherche d'un arbre couvrant minimal _____

Dans l'exemple ci-après d'un graphe à 6 sommets et 12 arêtes possibles totalisant 201 unités (fig. 10.8), la méthode de Kruskal forme un arbre couvrant avec 5 arêtes totalisant 54 unités (fig. 10.9).

coût	a	b	c	d	e	f
a	0	10	15	20	22	20
b		0	18	25		
c			0		11	9
d				0	20	17
e					0	14
f						0

⇒ tri ⇒

coût	L
9	c, f
10	a, b
11	*c, e*
14	e, f
15	a, c
17	d, f
18	b, c
20	a, d
20	a, f
20	d, e
22	a, e
25	b, d

⇒

R	Y	Z
c f	c, f	a, b, d, e
c **e**	+e	a, b, d
a c	+a	b, d
a **b**	+b	d
d f	+d	ϕ

Fig. 10.8 Méthode de Kruskal ; exemple de traitement.

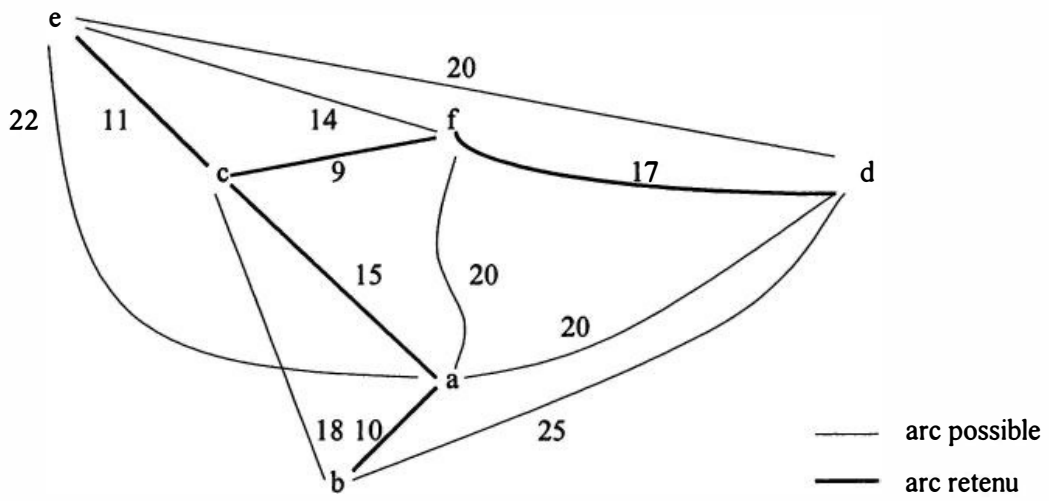


Fig. 10.9 Arbre couvrant minimal. Résultat.

10.9 POINTS, ENSEMBLES D'ARTICULATION

DÉFINITION 10.13 – Un ***point d'articulation*** est un sommet dont le retrait déconnecte une composante connexe ou un graphe connexe.

S'il faut enlever plusieurs points à la fois pour déconnecter le graphe, ces points forment un ***ensemble d'articulation.***

PROPOSITIONS

10.12 – Tout sommet non terminal d'un arbre est point d'articulation (comme centre d'une structure en étoile...).

10.13 – Un graphe qui admet un circuit hamiltonien n'a pas de point d'articulation.

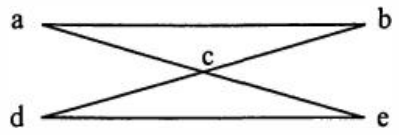


Fig. 10.10 Graphe « papillon » ; c, point d'articulation.

10.10 ISTHME

DÉFINITION 10.14 – Un ***isthme*** est une *arête* dont le retrait[1] déconnecte un graphe connexe ou une composante connexe.

PROPOSITIONS

10.14 – Toute arête d'un arbre est un isthme.

10.15 – Aucune arête d'un cycle ou d'un circuit n'est un isthme.

10.16 – Un graphe possédant un cycle eulérien n'a pas d'isthme.

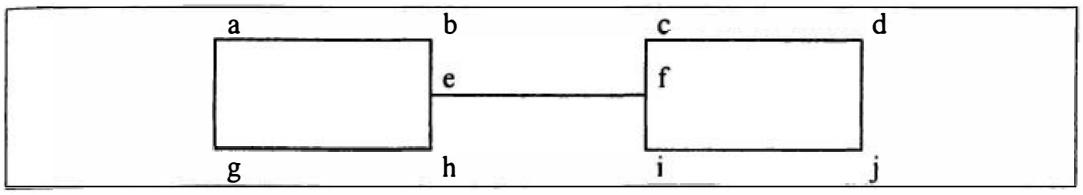


Fig. 10.11 Graphe «en haltère ». Isthme (e f) ; points d'articulation e, f.

10.11 K-CONNEXITÉ ET ROBUSTESSE STRUCTURELLE

DÉFINITION 10.15 – Un graphe connexe G est dit

- ***k-connexe*** s'il faut lui enlever *k sommets au moins* pour le déconnecter,
- ***k-arête-connexe*** s'il faut lui enlever *k arêtes au moins* pour le déconnecter.

PROPOSITIONS

10.17 – Une chaîne, un arbre, un graphe possèdant un isthme sont 1-connexe et 1-arête-connexe.

10.18 – Un cycle hamiltonien est 2-connexe et 2-arête-connexe.

Le ***papillon*** (fig. 10.10) est un graphe 1-connexe du fait du point d'articulation c mais 2-arête-connexe puisque chaque arête est sur un cycle.

L'***haltère*** (fig. 10.11) est un graphe 1-connexe du fait des points d'articulation e et f, et 1-arête-connexe du fait de l'isthme (e f).

[1] Le retrait ne porte que sur l'arête, pas sur les sommets aux extrémités.

On peut considérer ces nombres comme *indicateurs de robustesse*, par exemple pour un réseau vis-à-vis des catastrophes naturelles, accidents...

On peut également les envisager pour juger de l'*équité d'un tournoi simplifié*.

Par exemple, un cube, graphe 3-connexe et 3-arête-connexe, peut être considéré comme un modèle parfaitement symétrique de tournoi entre 8 personnes, avec 12 rencontres, chacune participant à 3 exactement sans avoir de rôle plus critique qu'une autre.

10.12 MULTIGRAPHES

DÉFINITION 10.16 – Un ***multigraphe*** est un graphe qui *possède plusieurs arcs ou arêtes de mêmes extrémités*. Il est dit ***p-graphe*** s'il n'a jamais plus de p arcs (ou arêtes) ayant mêmes extrémités (BERGE, 1983).

On le représente

- par une matrice d'adjacence (§ 7.8.1) quand une case (x, y) peut contenir une somme symbolique individualisant suffisamment chaque arc ou arête reliant x à y,
- plus souvent, par des matrices d'incidence sommet/arc (§ 7.8.3) dont chaque colonne représente un arc individualisé.

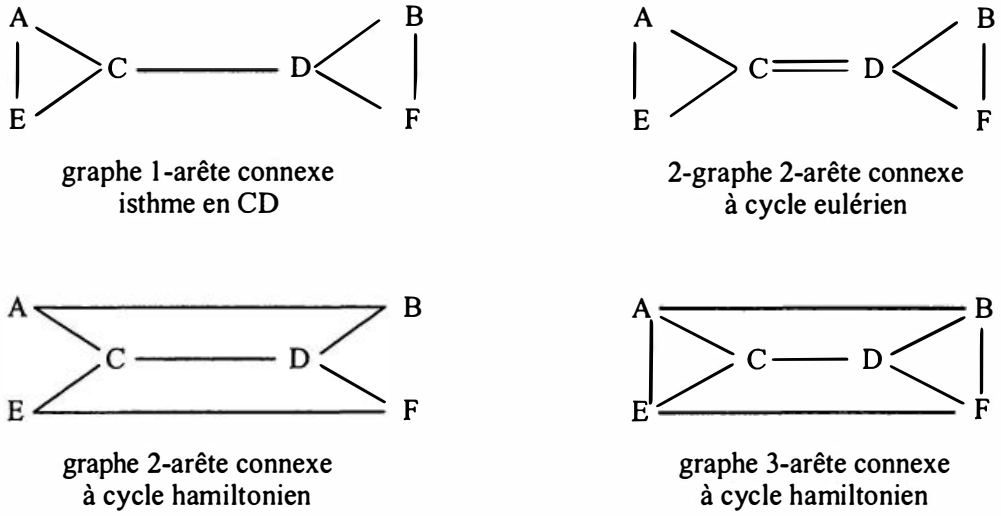


Fig. 10.12 Connexité aux arêtes.

EXEMPLES

- Une formule développée de *chimie organique* correspond à un multigraphe dès qu'on explicite des liaisons non saturées :
  - l'acétylène dénotée H–C≡C–H, (on parlera alors de 3-graphe),
  - le schéma de Kékulé pour les noyaux benzéniques, etc.
- Un *circuit électrique* correspond à un multigraphe quand plusieurs composants sont montés entre deux mêmes points.
- Un multigraphe modélise tout *réseau de communication ou de distribution* où existent des liaisons doubles ou multiples entre points, la robustesse du réseau face aux risques de rupture de liaison étant régie par l'arête-connexité du multigraphe.

- Intelligence artificielle / modélisation. Un multigraphe peut être utilisé pour *représenter simultanément plusieurs relations relatives à un même champ sémantique* : cette convention est à la base des réseaux sémantiques, pour lesquels des relations secondaires seront calculables par composition des relations fondamentales retenues.

10.12.1 Base de cycles

Considérons un (multi)graphe connexe G = <X, U>, et dans G un *cycle* c *élémentaire* parcouru dans un certain sens.

Supposons un sens et un numéro spécifique i associé à chaque arête u_i. Alors, au parcours de c on peut associer un t-uple $v = \{ v_i\}$, où chaque v_i vaut

- 1, si c emprunte u_i dans le sens prévu
- 0, si c n'utilise pas u_i,
- –1, si c utilise u_i à contresens.

– $v = \{-v_i\}$ décrit alors le même cycle que v, mais parcouru dans l'autre sens.

On peut alors introduire la somme voire la somme algébrique de cycles, les parcours multiples d'un même cycle…

PROPOSITIONS

10.19 – (GONDRAN) Tout cycle d'un multigraphe s'interprète comme somme de cycles élémentaires (identiques ou) disjoints.

10.20 – (BERGE) Les cycles d'un multigraphe forment un espace vectoriel.

10.21 – Les cycles d'un multigraphe s'expriment comme combinaisons linéaires d'un nombre réduit de cycles formant une ***base de cycles*** du multigraphe.

10.12.2 Nombre cyclomatique

DÉFINITION 10.17 – Soit un multigraphe G, avec n sommets, m arêtes, p composantes connexes. Posons :

$$\rho(G) = n - p$$

$$\nu(G) = m - \rho(G) = m - n + p$$

$\nu(G)$ est par définition le ***nombre cyclomatique*** de G.

PROPOSITIONS

10.22 – (BERGE) Soit G un multigraphe, et G^+ un multigraphe obtenu en ajoutant à G une arête (a, b), où a et b sont déjà sommets de G.

Si a et b sont confondus ou déjà liés par une chaîne dans G alors

$$\rho(G^+) = \rho(G) \; ; \; \nu(G^+) = \nu(G) + 1$$

sinon

$$\rho(G^+) = \rho(G) + 1 \; ; \; \nu(G^+) = \nu(G)$$

10.23 – (BERGE) Pour tout graphe G = <X, U> $\rho(G) \geq 0$ $\nu(G) \geq 0$.

PREUVE. On part d'un graphe G_0 = <X, ∅> ayant comme sommets les n sommets de G, mais 0 arête et donc n composantes connexes, pour lequel $\rho = 0$, $\nu = 0$. On lui ajoute une à une les arêtes de G, faisant croître (mais jamais décroître) soit ρ soit ν, qui ne peuvent donc être que positifs (ou nuls). ■

10.24 – (BERGE) Le *nombre cyclomatique* $\nu(G)$ d'un *multigraphe* G est égal au nombre maximum de cycles indépendants.

10.25 – (BERGE) G est sans cycle *ssi* $\nu(G) = 0$.

10.26 – (BERGE) G possède un cycle unique *ssi* $\nu(G) = 1$.

10.27 – (BERGE) Dans un graphe fortement connexe G, le nombre cyclomatique $\nu(G)$ est égal au nombre maximum de *circuits* indépendants.

10.12.3 Applications

Electricité /Electronique

Considérons un *schéma électrique*, assimilable à un multigraphe, sur la base d'un composant (résistance, inductance, condensateur, générateur) par arête, et d'un sommet par nœud de connexion.

La mise en équation peut se faire en considérant un courant par branche (composant ou suite de composants montés en série). Dans le cas de la figure 10.13 ci-après, nous aurions 37 courants de branche inconnus, liés par

- 22 équations de Kirchoff aux nœuds, portant sur les courants,
- 16 équations de Kirchoff aux mailles, portant sur les tensions relatives aux branches.

De préférence, la mise en équation se fait selon la méthode des *courants fictifs de Maxwell* : on introduit un courant bouclé sur lui-même pour un premier circuit élémentaire, puis pour un second circuit utilisant au moins une branche sans courant affecté etc., jusqu'à ce que chaque branche soit traversée par au moins un de ces courants.

Les circuits sont les mailles auxquelles nous appliquons la loi de Kirchoff aux tensions. Soit ici 16 mailles ou circuits.

Les 16 courants inconnus correspondants fournissent

- au signe près, 16 courants de branche,
- les 21 autres courants de branche par des sommes algébriques du style $\Sigma s_k i_k$ avec $1 \leq k \leq 16$ et $s_k \in \{-1, 0, +1\}$.

Le nombre de courants fictifs de Maxwell est précisément fixé par le nombre cyclomatique du multigraphe de connexion, à savoir : ν = branches – nœuds + 1 ; ici $\nu = 37 - 22 + 1 = 16$.

NOTE. Comme l'a montré Yves Rocard, tout résultat applicable en électricité se transpose immédiatement à la mécanique des vibrations, l'acoustique, la thermique et, potentiellement, à bien d'autres domaines dans lesquels on développera des interprétations physiques diverses, mais analogues, d'un même modèle de ***système linéaire.***

		1→	2→	3→	4→	
		↓	5	6	7	↓ 4
			←8	←9	←10	
		↓	11	12	13	↓14
			15	16	17	
1↑		↓	18	19	20	↓21
			22	23	24	
		↓	25	26	27	↓28
			29	30	31	
		↓	32	33	34	↓35
			←37	←36	←35	

courants de branches + équations aux nœuds :
37 équations à 37 inconnues

	(1)	(2)	(3)
	(4)	(5)	(6)
↑16 ↓	(7)	(8)	(9)
	(10)	(11)	(12)
	(13)	(14)	(15)

courants fictifs de Maxwell :
16 équations à 16 inconnues
+ 21 définitions auxiliaires

Fig. 10.13 Les courants fictifs de Maxwell comme bases de cycles.

Génie logiciel

Pour améliorer testabilité et maintenabilité des logiciels, McCabe (1976) a suggéré d'employer le nombre cyclomatique du « graphe de structure » d'un traitement comme mesure de sa complexité structurelle (ou décisionnelle), largement adopté depuis [Watson & McCabe].

Dans cet esprit, tout sommet représente une instruction, et tout arc une transition entre instructions.

La complexité est alors donnée par

$$\nu(G) = \text{nbTransitions} - \text{nbSommets} + \text{nbComposantes}$$

Une suite d'instructions ou séquence est assimilable à une instruction unique, car cette agrégation supprime autant de sommets que d'arcs, et laisse $\nu(G)$ invariant.

Un sommet réalise une confluence de traitements si son entrance excède 1, une divergence (ou étape décisionnelle) si sa sortance dépasse 1.

Les divergences correspondront aux *if, until* ou *while* terminaux, *case, alt, par* ; les convergences, aux cibles de branchements, aux *repeat* ; tandis que les *while* ou *until* initiaux seront mixtes.

```
100  input « x= » ; x
110  input « y= » ; y
120  if x>y then 300
130  if x=y then 200
140  print x, « plus petit que », y
150  goto 999
200  print x, « égale », y
210  goto 999
300  print x, « plus grand que », y
999  stop
```

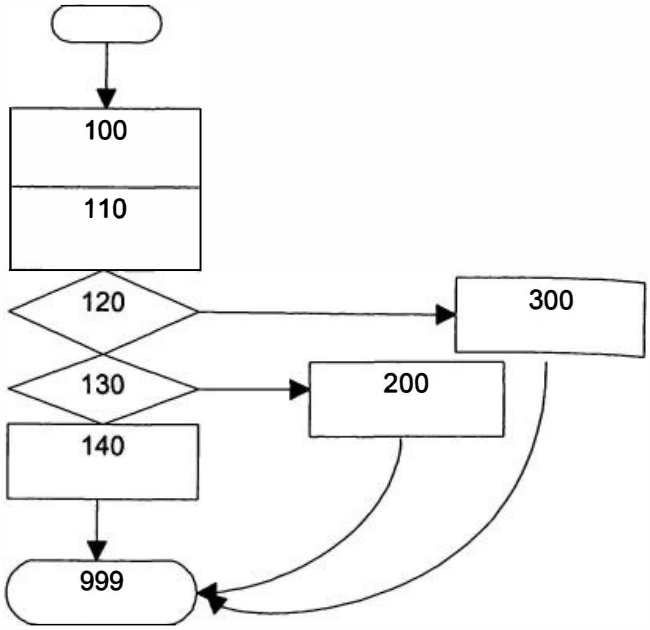


Fig. 10.14 Un programme BASIC : énoncé et graphe de structure.

NOTE. La mesure calculée ne change pas si 140/150 et/ou 200/210 sont remplacés par un seul nœud, car on ôte à la fois un arc et un nœud. Elle reste égale au nombre de décisions binaires.

10.13 EXERCICES

10.13.1 (Berge)

En 1870, le premier réseau de communications par pigeon-voyageurs assurait les liaisons données ci-après.

Chercher une/les *composante(s) fortement connexe(s)* et définir le statut de chaque ville au regard du réseau.

de à	Bordeaux	Grenoble	Lyon	Marseille	Nice	Paris
Bordeaux						x
Grenoble					x	
Lyon	x	x		x		
Marseille			x		x	
Nice						
Paris		x	x			

10.13.2

Montrer que dans tout graphe symétrique le diamètre est au plus égal au double du rayon.

10.13.3

On considère un logiciel, ses modules, ses procédures, leurs appels respectifs.

Le graphe d'appel des procédures forme-t-il une hiérarchie ?

Et le graphe d'appel des modules ?

10.14 PROBLÈMES

10.14.1 Méthode du palmier

Discuter la proposition suivante.

Pour tracer aussi clairement que possible les graphes connexes ayant de nombreux circuits (et n'admettant donc pas de fonction ordinale) :

- chercher les circuits élémentaires du graphe et les classer par taille décroissante,
- organiser la représentation à partir du circuit élémentaire le plus long, puis du suivant etc. dans l'ordre des tailles décroissantes.

10.14.2

On considère une arborescence A. On appelle hauteur de chacun de ses sommets x :

$$h(x) \equiv (\Gamma^{-1}(x) = \varnothing \rightarrow 0 \; ; \; 1 + h(\Gamma^{-1}(x)))$$

Qu'appellera-t-on hauteur de l'arbre ?

Peut-on s'en servir pour mesurer la complexité d'un questionnaire, d'un menu, d'un système arborescent de menus ?

10.14.3

Dans un arbre ou une arborescence, on appelle distance d(x, y) entre sommets x et y la taille de la chaîne élémentaire allant de l'un à l'autre. Vérifier le bien-fondé de cette définition.

On considère un système arborescent de menus et sous-menus. Que représente d(x, y) dans cette arborescence ? Que se passe-t-il si y (ou l'action attachée) doit succéder fréquemment à x (ou l'action attachée) alors que d(x, y) est grand ? Impact sur les règles de construction des menus.

10.14.4

En 1985, le responsable de la Centrale de Traduction Automatique de l'Union Européenne, centrale établie à Luxembourg, déclarait : « Nous traduisons automatiquement le français en anglais, et réciproquement ; notre prochain objectif, bientôt réalisé, nous permettra de traduire automatiquement toute langue germanique [officielle de l'UE] en anglais, et toute langue romane [officielle de l'UE] en français ».

Etait-il encore loin de l'objectif « traduire toute langue officielle de l'UE dans n'importe quelle autre langue officielle de l'UE » ? (Chiffrer en %.)

Tracer le graphe des traductions possibles. Est-il connexe ? fortement connexe ? Donner les conséquences pratiques.

10.14.5

Votre oncle Marmaduke Gascoyne, officier retraité de l'Armée des Indes, prétend qu'en matière ferroviaire, il préfère un réseau hamiltonien à un réseau à la Kruskal. Pourquoi ?

10.15 PISTES DE RÉFLEXION

10.15.1

Expliciter sous quelles conventions une formule de chimie développée peut être assimilée à un graphe.

Que deviennent les atomes, les valences ? Cas des liaisons non saturées.

Question des isomères, des polymères (Polyà, 1936).

10.15.2

Considérer un réseau routier ou ferroviaire. Que signifie pour une ville

- un écartement élevé ?
- être un centre pour un tel réseau ?

Y a-t-il intérêt à y implanter une usine ? un entrepôt ?

Applications :

- A partir d'une table des distances ville à ville, montrer que le centre de la France, au sens de l'actuel réseau routier, se situe entre Bourges et Orléans.
- Implantation de la capitale d'un pays neuf.

10.15.3

Discuter le concept d'arbre ou d'arborescence appliqué

- à la structuration d'un document comme ce texte,
- à la classification de documents (par exemple, classification décimale universelle des bibliothécaires et documentalistes).

10.15.4

Discuter le concept d'arbre ou d'arborescence appliqué à la production de formulaires à partir de textes d'arrêtés ou de règlements.

10.15.5

Soit un réseau routier à concevoir, pour un budget limité.

Comment tenir compte

- des coûts (supposés pour une liaison (x y) proportionnels à sa longueur),
- de l'intérêt – estimé pour la même liaison (x y) par un prédicteur de fréquentation :

$$\text{population}(x) \cdot \text{population}(y) \,/\, \text{distance}^2(x, y) \ ?$$

10.16 VOCABULAIRE

On trouvera un glossaire trilingue plus étendu dans (BERGE, 1983).

cas orienté		cas symétrique	
graphe (orienté)	*digraph, directed graph*	graphe symétrique, non orienté	*(symmetric) graph*
sommet	*vertex (vertices)*	sommet	*vertex (vertices)*
sommet initial, terminal	*root, ending vertex*	sommet pendant	*pendant vertex*
prédécesseur, successeur	*predecessor successor*	sommet adjacent	*neighbour*
arc	*directed edge*	arête	*edge*
chemin	*directed path*	chaîne	*path*
boucle	*(1-)loop*	boucle	
circuit (de taille k)	*circuit, (k-)loop*	cycle	*cycle*
graphe (fortement) connexe	*(strongly) connected graph*	graphe connexe	*connected graph*
graphe sans circuit hiérarchie	*directed acyclic graph, DAG*		
arborescence (graphe orienté sans cycle, à sommet initial unique)	*directed tree*	arbre, graphe sans cycle	*tree*
accessibilité	*reachability*		
		point d'articulation	*cutpoint*
		ensemble d'articulation	*cutset*
		isthme	*bridge*

10.17 LECTURES

BERGE C., *Théorie des graphes et ses applications*, Dunod, CUM n° 2, 1963.

BERGE C., *Graphes*, Gauthier-Villars, 1983.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien*, tome I, chap. VI, Coll. Programmation, Ed. Gauthier-Villars, 1973.

GONDRAN M. & MINOUX M., *Graphes et algorithmes*, chap. III, collection EDF n° 37, Eyrolles, 1979-95...

KUNTZMANN J., *Théorie des réseaux*, Dunod, 1972.

MCCABE T.J., A Complexity Measure, *IEEE Transactions on Software Technology*, vol SE-2, n° 4, dec. 1976, pp. 308-320.

PAIR C., DERNIAME J.C., *Problèmes de cheminement dans les graphes*, Monographie Afcet, Dunod, 1972.

PICARD C.F., *Théorie des questionnaires*, Gauthier-Villars, 1965.

PICARD C.F., *Graphes & questionnaires*, Gauthier-Villars, coll. Programmation, 1972 (2 vol.).

PRATHER R.E., *Discrete Mathematical Structures for Computer Science,* Houghton Mifflin C^y, Boston, 1976.

ROBILLARD P.N. *Le logiciel, de sa conception à sa maintenance,* Ed. Gaëtan Morin, Chicoutimi, Québec, 1985 (§ 10.5 : complexité selon McCabe).

ROY B., *Algèbre moderne et théorie des graphes*, tome 2, Dunod, 1970.

VÉLU J., *Méthodes mathématiques pour l'informatique,* Dunod, 1994, chap. 20.

WATSON A.H., MCCABE T.J., *Structured Testing : a testing methodology using the Cyclomatic Complexity Metric,* NIST special publication 500-235, 1996[2].

coll., *Petite encyclopédie des mathématiques,* chap. XXXVI, Edition Pagoulatos, Didier distr.

[2] Disponible sur Internet.

CHAPITRE 11

GRAPHES MULTIPARTIS

Pour modéliser et résoudre les problèmes d'exclusion mutuelle, traiter certains jeux, et pour introduire aux hypergraphes.

11.1 PRÉSENTATION

DÉFINITION 11.1 – Un graphe G = <X, U> est dit ***graphe k-parti***[1] s'il existe une partition $\{X_1, X_2 \dots X_k\}$ de X formée de k sous-ensembles non vides, et tels que

$$\forall (x, y) \in U \; \exists \, i, j \; (x \in X_i \wedge y \in X_j \wedge i \neq j)$$

On note alors $G = <(X_1, X_2 \dots X_k) ; U>$.
On parlera de ***graphe multiparti*** si on ne désire pas préciser k.

Le tracé de tels graphes est souvent facilité en regroupant les sommets d'une même classe.

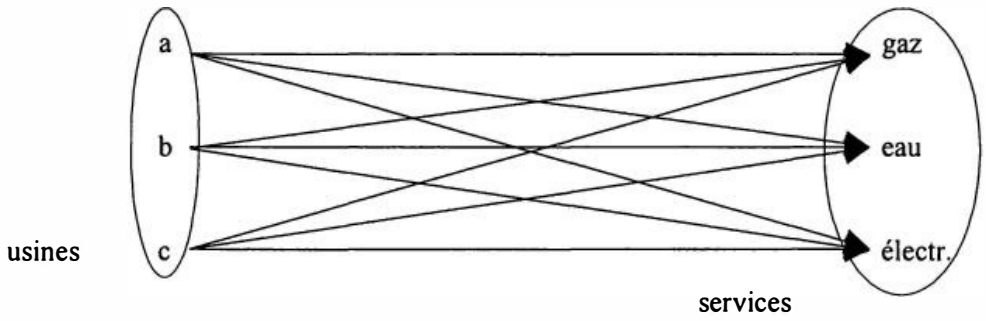


Fig. 11.1 Graphe biparti $K_{3,3}$.

Si l'ensemble X est homogène, la relation associée à G est une *relation d'incompatibilité* ; alors $\{X_1, X_2 \dots X_k\}$ est une partition de X respectant cette incompatibilité. C'est pourquoi on dit que les X_i forment des *classes de compatibilité.*

EXEMPLES

- X est un ensemble de personnes, les X_i des regroupements respectant les *incompatibilités d'humeur.*
- Soit un graphe figurant les *flux de matière* entre 4 groupes d'agents économiques, X_1 étant un ensemble de fournisseurs de matières premières, X_2 un ensemble d'usines, X_3 un ensemble de centres de distribution, X_4 l'ensemble des clients finaux. *Ce graphe est 4-parti ssi les éventuels flux entre membres d'une même*

---

[1] Parti au sens ancien de « partagé » comme dans habit parti, maille à partir, partition...

classe restent indirects i.e. médiatisés par (des agents) d'autres classes (ex : usine$_1$ → distributeur$_k$ → usine$_2$).

11.2 TESTS DE MULTIPARTISME

Un graphe donné ne serait-il pas *de fait* un graphe k-parti, et pour quel k ?

PROPOSITIONS

11.1 – Tout graphe k-parti est nécessairement irréflexif (pour k = 1, le graphe est même sans arc).

11.2 – Le graphe d'une ordre total strict, ou ***échelle***, transitif et irréflexif, est sur un ensemble fini un graphe N-parti, où N est le nombre total de points.

11.2.1 Bipartisme

Les tests pour k = 2 (bipartisme) cherchent souvent à prouver que *tout cycle est de taille paire*, propriété dont la nécessité s'établit par l'absurde.

11.2.3 Fonction de Grundy

Présentation

DÉFINITION 11.2 – Une fonction de Grundy d'un graphe G = <X, Γ> associe à chaque sommet x de G un nombre naturel g(x) qui est le plus petit entier qui n'est pas associé par g à un successeur de x :

$$g(x) = \min (\mathbb{N}_0 - \{g(y) \mid y \in \Gamma x\})$$

PROPOSITIONS

11.3 – Pour que g existe, G doit être irréflexif.

11.4 – Dans un graphe orienté, si x est un sommet terminal alors g(x) = 0.

11.5 – Si un graphe G = <X, U> possède une fonction de Grundy g, et si X_n est le sous-ensemble de tous les sommets x ∈ X tels que g(x) = n, les X_i partitionnent X en k classes, et G peut être vu comme un graphe k-parti G = <(X_0,X_1, … X_{k-1}) ; U>.

NOTE. Ici on ne teste pas si le graphe est k-parti pour une valeur de k connue : au contraire, l'éventuelle fonction de Grundy propose une valeur de k possible pour le graphe, en dehors de toute hypothèse particulière.

Calcul informel de la fonction de Grundy d'un graphe orienté

- On marque par 0 tout sommet terminal du graphe (à défaut, on marque 0 des sommets non adjacents).
- Pour les sommets non terminaux dont tous les successeurs sont marqués : si les successeurs du sommet s sont respectivement marqués i, j, k…, s doit être marqué par *le plus petit entier*, 0 compris, *absent de la liste des i, j k, …*
- A défaut, on marque provisoirement ≠ 0 tous les sommets ayant au moins un successeur à 0 ; et 0 les sommets dont tous les successeurs sont marqués ≠ 0.

fin calcul de la fonction de Grundy

Il est clair que la fonction de Grundy du graphe existe si la phase 2 se termine, ce qui se produit rapidement s'il s'agit d'un arbre ou d'une hiérarchie (fig. 11.2).

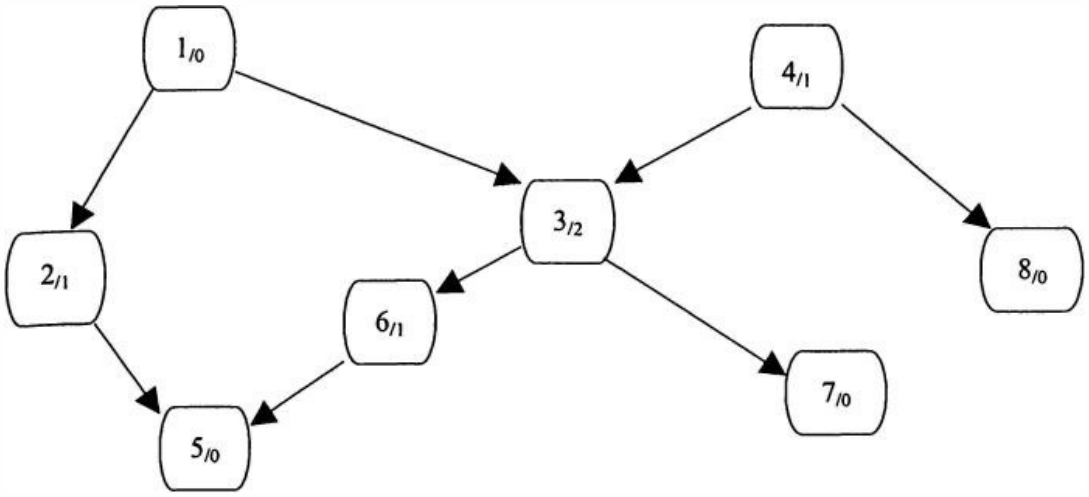


Fig. 11.2 Fonction de Grundy d'une hiérarchie.

Successivement :

- 5, 7, 8 sont marqués 0 comme terminaux,
- 2 et 6 sont marqués 1 (leurs successeurs sont terminaux),
- 3 est marqué 2 (successeurs marqués 1(6) et 0 (7)),
- 1 est marqué 0 car ni 2 ni 3 sont marqués 0,
- 4 est marqué 1 car 8 est marqué 0, et 3 n'est pas marqué 1.

En cas de circuit, on tente de développer un marquage cohérent en assignant à un sommet indéterminé un marquage compatible avec les *interdits* connus.

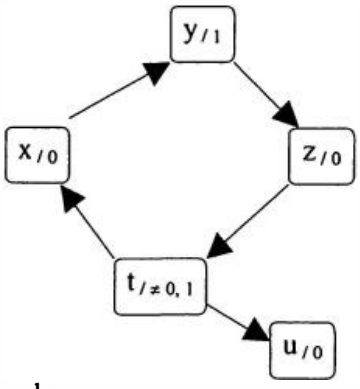
EXEMPLE. Soit, ci-après, un graphe G = <{x, y, z, t, u}, {xy, yz, zt, tu, tx}> comportant un circuit de taille 4 (x y z t) et un arc terminal (t u).

Immédiatement, f(u) = 0 ; par suite, f(t) ≠ 0 ; donc f(z) = 0, puisque t est seul successeur de z ; et de là, f(y) = 1, f(x) = 0, et finalement f(t) = min($\mathbb{N}_0$ – {0, 0}) = 1 (fig. 11.3).

L'arc terminal a introduit une dissymétrie suffisante pour rendre la solution unique.

G peut finalement être considéré comme biparti, avec X_0 = {x, z, u} et X_1 = {y, t}.

Si le circuit avait été de taille impaire, on aurait obtenu un graphe triparti.



graphe

X	Γ	passes 1					2
x	y					0	
y	z				1		
z	t			0			
t	x, u		≠ 0				1
u	∅	0					

calcul de la fonction de Grundy

Fig. 11.3 Fonction de Grundy d'un graphe avec circuit.

Problèmes d'existence et d'unicité

Pour les *graphes irréflexifs,* une fonction de Grundy existe *dans la plupart des cas* (1-3) *sans être nécessairement unique* (5). On a à ce sujet divers énoncés.

PROPOSITIONS

11.6 – (BERGE) Tout graphe symétrique admet une fonction de Grundy.

11.7 – (BERGE) Tout graphe transitif admet une fonction de Grundy.

11.8 – (BERGE) Tout graphe sans *circuit de taille impaire* admet une fonction de Grundy.

11.9 – (BERGE) Un circuit *isolé* de taille impaire n'admet pas de fonction de Grundy.

11.10 – (BERGE) Un circuit *isolé* de taille paire possède 2 fonctions de Grundy.

11.11 – (GRUNDY, 1939) Un graphe sans circuit admet une fonction g unique. En tout point, g(x) est au plus égal à la taille du plus long chemin (élémentaire) issu de x.

Notion de noyau

Dans certains problèmes, où les sommets représentent des *situations* et les arcs des *actions*, on s'intéresse à l'*ensemble des situations stables, ou* ***noyau,*** ainsi défini :

DÉFINITION 11.3 – On appelle noyau N d'un graphe $G = \langle X, \Gamma \rangle$ un sous-ensemble de sommets $N \subseteq X$ tel que :

$$\forall\, x \in N, \Gamma x \cap N = \varnothing, \text{ et } \forall\, x \in X - N, \Gamma x \cap N \neq \varnothing$$

Ainsi, si je me trouve hors du noyau, je peux y revenir immédiatement ; mais étant dans le noyau, ou la situation est terminale, ou je sors du noyau.

PROPOSITION 11.12 – Pour un graphe possédant une fonction de Grundy, l'ensemble des sommets de marque 0 forme un noyau.

NOTE. La partition des sommets d'un graphe en un noyau N et son complément X–N ne coïncide pas avec la notion de graphe biparti, car les sommets de X–N peuvent avoir un ou plusieurs successeurs dans X–N dès lors qu'ils en ont un dans N. Cependant, la proposition 11.5 s'applique, avec $N = X_0$.

11.3 JEUX & FONCTIONS DE GRUNDY

11.3.1 Principe

Soit un graphe modélisant un jeu simple, à 2 joueurs, à raison d'un sommet par situation, et d'un arc par action possible.

Supposons de plus toute ***situation terminale*** *gagnante pour celui qui l'atteint.* Alors le noyau forme l'ensemble des positions non perdantes. En effet, si j'atteins une position du noyau :

- ou elle est terminale, et j'ai gagné,
- ou elle n'est pas terminale, mon adversaire jouera ce qu'il veut mais sortira du noyau, et je pourrais revenir au noyau au coup suivant.

Cette façon de faire est *non perdante* : elle est *gagnante sauf si on boucle*, i.e. gagnante si le jeu se réduit nécessairement à un nombre fini de mouvements.

EXEMPLE. (Jeu de Nim) 2 joueurs ont devant eux un tas de 37 allumettes. Chacun à son tour en prend de 1 à 4. Celui qui prend la dernière a gagné.

La fonction de Grundy attachée ici à un tas de n allumettes est g(n) = (n mod 5). *Partant de 37 allumettes, si le premier joueur en retire 2, il a potentiellement gagné.* A 35 allumettes correspond une marque 0.

Le second joueur jouant k allumettes, il suffit au premier de contrer en en retirant 5 – k pour revenir à 30, 25, 20, 15, 10, 5 puis 0 allumette, et ainsi gagner.

	après prise / mouvement				fonction de Grundy
avant	1	2	3	4	
0	/	/	/	/	0
1	0	/	/	/	1
2	1	0	/	/	2
3	2	1	0	/	3
4	3	2	1	0	4
5	4	3	2	1	0
6	5	4	3	2	1
7	6	5	4	3	2
8	7	6	5	4	3

Fig. 11.4 Analyse d'un Jeu de Nim.

Plus généralement, soit 2 joueurs ayant devant eux un tas de n objets. Si chacun à son tour en prend de 1 à k, avec la règle que « celui qui prend la dernière a gagné », alors la fonction de Grundy attachée ici au tas de n objets est g(n) = n mod (k + 1).

PREUVE. On voit d'abord que $g(n) \equiv n$ si $0 \leq n \leq k$. D'où $g(k + 1) = 0$ par définition, et par récurrence $g(n) \equiv n \bmod (k + 1)$. ■

Par conséquent,

- si un joueur reçoit un nombre d'objets congru à r modulo k + 1,où r > 0, il en retire r et revient au noyau : soit il reste 0 objet, et il a gagné, soit l'adversaire hérite d'un multiple de k + 1 ;
- si un joueur A part d'un multiple de k + 1 en retirant $1 \leq p \leq k$ objets, l'autre joueur B peut le contrer en retirant m = k + 1 – p objets (où $1 \leq m \leq k$), d'où un nouveau multiple de k + 1 : si c'est 0, B est victorieux, sinon B remet A dans une situation semblable à la précédente ; le nombre d'objets restant décroissant de k + 1 d'une fois sur l'autre, B est ainsi sûr de gagner.

11.3.2 Composition

Un jeu J peut parfois être considéré comme composition de sous-jeux indépendants $\{j_1, j_2, j_3 \ldots j_n\}$, et la situation globale dans J est alors le n-uple des situations particulières dans les sous-jeux. Si g_1, g_2, g_3, g_n sont les fonctions de Grundy relatives aux sous-jeux, comment bâtir la fonction de Grundy G associée au jeu global J ?

BERGE (63, 83) propose deux compositions liées à deux styles d'évolution :

- si chaque joueur joue *à chaque fois dans chacun* des jeux, toutes les composantes du n-uple peuvent évoluer à la fois, J sera dit ***produit cartésien*** des j_i, ce qu'on notera $J = j_1 \otimes j_2 \otimes j_3 \otimes \ldots \otimes j_n$;
- si chaque joueur joue *à chaque fois dans un seul* des jeux, une seule composante du n-uple évolue à la fois, J sera dit ***somme cartésienne*** des j_i, ce qu'on notera $J = j_1 \oplus j_2 \oplus j_3 \oplus \ldots \oplus j_n$.

Seule la seconde piste a mené à des résultats exploitables.

DÉFINITION 11.4 – La ***somme digitale*** de deux entiers n_1 et n_2 est l'entier $n = n_1 \oplus n_2$ ainsi formé[2] : considérons la forme binaire de n, n1, n2 ; alors, partant de

$$n_1 \equiv \Sigma\, bit(n_1, k)2^k\,,\ n_2 \equiv \Sigma\, bit(n_2, k)2^k$$

on obtient $n \equiv \Sigma\, bit(n, k)2^k$ si $bit(n, k) = (bit(n1, k) + bit(n2, k)) \bmod 2 = bit(n1, k) \oplus bit(n2, k))$.

LEMME 11.13 – Pour tout entier n, $n \oplus n \equiv 0$.

PROPOSITION 11.14 – La somme digitale dote l'ensemble $\mathbb{N}$ d'une structure de groupe.

PROPOSITION 11.15 – La fonction de Grundy G de la somme cartésienne J de graphes j_i possédant une fonction de Grundy g_i est la somme digitale de ces fonctions de Grundy :

$$G(j_1 \oplus j_2 \oplus j_3 \oplus \ldots \oplus j_n) = g_1 \oplus g_2 \oplus g_3 \oplus \ldots \oplus g_n$$

EXEMPLE. On considère un jeu formé de 4 tas de p, q, r, s allumettes respectivement, et on décide qu'à chaque fois, le joueur

- choisit un tas,
- duquel il retire de 1 à 4 allumettes.

Gagne celui qui retire la dernière allumette du dernier tas.

Supposons qu'ici on ait initialement 13, 15, 17 et 19 allumettes.

Les fonctions de Grundy de chaque tas sont ici son nombre d'allumettes modulo 5 : respectivement 3, 0, 2, 4, et la fonction de Grundy globale est ici $3 \oplus 0 \oplus 2 \oplus 4 = 011 \oplus 000 \oplus 010 \oplus 100 = 101$. Pour abaisser cette valeur à 0 (coup gagnant) on a le choix entre

- $g_1 = 011 \rightarrow 011 \oplus 101 = 110$, qui n'est pas une valeur possible pour g_1 (6>4),
- $g_2 = 000 \rightarrow 000 \oplus 101 = 101$, qui n'est pas une valeur possible pour g_2 (5>4),
- $g_3 = 010 \rightarrow 010 \oplus 101 = 111$, qui n'est pas une valeur possible pour g_3 (7>4),
- $g_4 = 100 \rightarrow 100 \oplus 101 = 001$, qui consiste à ramener 19 ($\equiv 4 \bmod 5$), à 16 ($\equiv 1 \bmod 5$) en retirant 3 ($1 \leq 3 \leq 4$) allumettes.

Si l'adversaire ramène maintenant le nouveau jeu de (13, 15, 17, 16) à (13, 15, 13, 16), la nouvelle fonction de Grundy globale est $3 \oplus 0 \oplus 3 \oplus 1 = 1$, et nous revenons au noyau en ramenant g_1 à 2 ($13 \rightarrow 12$) ou g_2 à 1 ($15 \rightarrow 11$) ou g_3 à 2 ($13 \rightarrow 12$) ou g_4 à 0 ($16 \rightarrow 15$). Si on se ramène à (12, 15, 13, 16) par exemple, la fonction de Grundy globale est alors $2 \oplus 0 \oplus 3 \oplus 1 = 10 \oplus 00 \oplus 11 \oplus 01 = 0$...

ATTENTION. Pour simplifier l'emploi des fonctions de Grundy pour la seule détermination de noyaux élémentaires, il arrive qu'on en exploite une version booléenne ; mais le principe de composition n'est pas applicable à ces pseudo-fonctions de Grundy.

11.3.3 Extension ludique

En vue de modéliser des jeux (ou des négociations), comment modifier la définition des fonctions de Grundy pour traiter aussi bien les ***sommets terminaux gagnants*** que les ***sommets terminaux perdants,*** ou ***pièges*** (fatals) ?

Assimilons ces sommets perdants à des sommets menant nécessairement à un sommet virtuel gagnant pour l'adversaire ; ce sommet virtuel serait coté 0, et son prédécesseur l'admettant comme seul successeur, 1. Nous poserons donc :

---

[2] Les informaticiens parleraient de ouExclusif *bit à bit* ou *entre mots*.

DÉFINITION 11.5 – La fonction de Grundy étendue aux jeux à pièges vaut

- 0 pour les sommets terminaux *gagnants,*
- 1 pour les sommets terminaux *perdants* ou pièges,
- le plus petit entier non affecté à ses successeurs pour tout autre sommet.

Cette symétrisation gagnant/perdant permettra de passer d'un jeu à son Qui-Perd-Gagne, par exemple du jeu dit « Fan Tan » à celui de « l'Année Dernière à Marienbad ». Ces jeux opposés ne diffèrent que par le marquage et non par la structure, ils peuvent ainsi avoir des débuts de partie similaires…

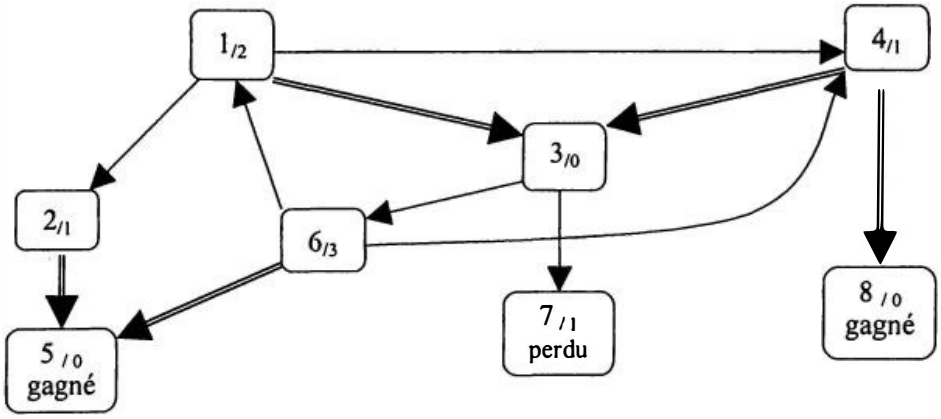


Fig. 11.5 Jeu à piège ; flèche double pour les coups gagnants.

NOTE. La nécessité d'un graphe irréflexif signifiera que, dans les jeux modélisés, passer son tour un nombre indéfini de fois devra être interdit, passer son tour un nombre limité de fois se traitant par réplication de tout ou partie du graphe.

11.4 COLORIAGE DE GRAPHES

Soit $G = <(X_1, X_2 \ldots X_k), U>$ un graphe k-parti. En associant une couleur différente à chacun des X_i, on peut considérer G comme un *graphe aux sommets coloriés*, 2 sommets adjacents du graphe étant toujours de couleurs différentes.

11.4.1 Graphe p-chromatique

DÉFINITION 11.6 – Un graphe sera dit ***graphe p-chromatique*** si on peut colorier ses sommets à l'aide de p couleurs distinctes, sans que 2 sommets adjacents soient de la même couleur.

Sur la base de cette coloration, on peut le considérer comme p-parti.

11.4.2 Nombre chromatique d'un graphe

DÉFINITION 11.7 – Le ***nombre chromatique*** $\gamma(G)$ d'un graphe G est *le plus petit p pour lequel il est p-chromatique* : c'est le *nombre minimal de couleurs nécessaires pour le colorier.*

PROPOSITION 11.16 – Pour un graphe G à n sommets, $1 \leq \gamma(G) \leq n$.

NOTE. La borne inférieure est donnée par le graphe dont chaque sommet est isolé, la borne supérieure par le graphe complet.

PROPOSITION 11.17 – Pour un graphe G contenant une clique K_n, $n \leq \gamma(G)$.

PROPOSITION 11.18 – Pour un graphe G k-parti, $\gamma(G) \leq k$.

Soit G = <(X_1,X_2,X_3,X_4,X_5) ;U> un graphe 5-parti ou 5-chromatique, tel que U ne comporte que des arcs allant de X_1 à X_2, X_2 à X_3, X_3 à X_4 et X_5, X_4 à X_1.

On peut lui associer un graphe-quotient g = <x, u> à raison d'un sommet x_i par classe X_i de G. Ce graphe g, similaire à celui vu en 4.2.3.1, possède une fonction de Grundy à 2 valeurs : *g et G sont de nombre chromatique 2* . En effet, si dans G je réunis X_1 et X_3 d'une part, X_2, X_4, et X_5 d'autre part, j'obtiens un graphe biparti, un coloriage de G qui respecte cependant toutes les incompatibilités de U ; et il est évident qu'une seule couleur ne suffit pas. Donc $\gamma(G) = 2$.

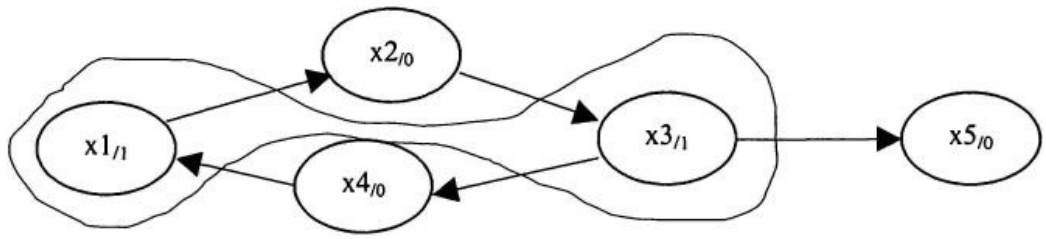


Fig. 11.6 Graphe 5-parti 2-chromatique.

11.4.3 Problème des 4 couleurs

Un imprimeur doit imprimer une carte de géographie, de telle façon que 2 pays ayant une frontière commune soient de couleurs différentes.

On peut procéder au coloriage ainsi. On construit un graphe symétrique, sur la base d'un sommet par pays[3] et, entre sommets, d'une arête *ssi* les deux pays correspondant aux sommets ont une frontière commune.

Dans le style des fonctions de Grundy, on marque

- 0 les sommets pendants[4],
- 1 les sommets qui ne sont liés qu'à des sommets 0,
- k les sommets qui ne sont liés qu'à des sommets de marque autre que k.

PROPOSITION 11.19 – Il suffit de 4 couleurs distinctes pour colorier une carte usuelle, supposée tracée sur une surface plane ou sphérique[5].

---

[3] Ou par *région* ou *face* ainsi définie : partie du plan ou de la sphère enclose par un cycle, et n'en contenant aucune autre.

[4] Dans un graphe symétrique, on appelle *sommet pendant* ceux qui ne possèdent qu'un sommet adjacent (ou voisin) ; pour colorier une carte d'Europe, ce seraient les sommets associés aux pays n'ayant qu'un voisin : Portugal, Monaco, Saint-Marin ...

[5] Le graphe associé est alors dit *planaire* car on peut le tracer dans un plan sans que 2 arcs ou arêtes se coupent ; la même propriété est valable sur une sphère, plan et surface de la sphère pouvant être mis en correspondance par une involution par rapport à un point de la surface de la sphère n'appartenant pas au tracé du graphe.

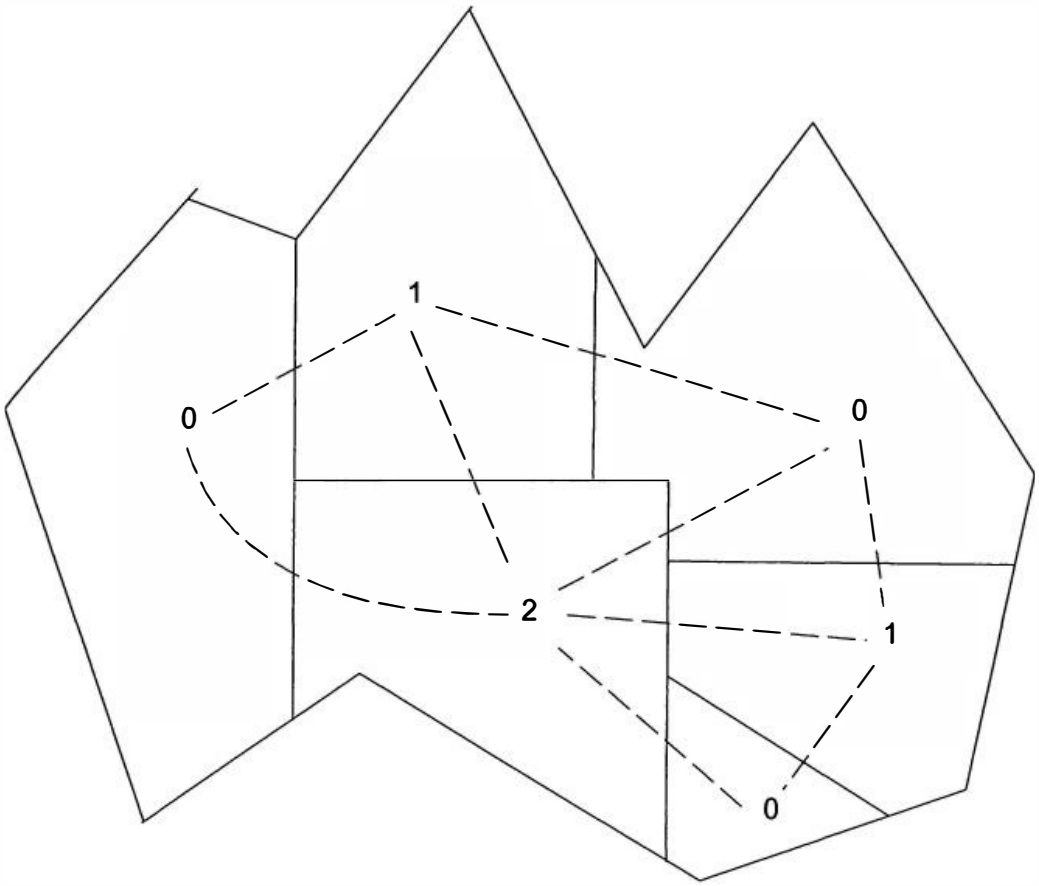


Fig. 11.7 Coloriage d'une carte avec 3 couleurs.

PREUVE. La proposition ci-dessus est restée longtemps conjecturale. K. Appel et W. Haken en ont donné en 1976 une démonstration par voie automatique, qui réduisait d'abord tous les cas possibles à 1509 cas irréductibles, puis vérifiait la conjecture pour chacun d'eux. L'impossibilité pratique de vérifier manuellement ce résultat l'a d'abord rendu peu crédible, mais il a depuis été corroboré plusieurs fois.

NOTE. Il faut 4 couleurs s'il y a un pays à 3 voisins adjacents 2 à 2, genre « autour du Luxembourg » : le graphe construit à raison d'un arc par pays et d'une arête par frontière contient alors une clique K_4, d'où $\gamma(G) \geq 4$ par la proposition 11.17.

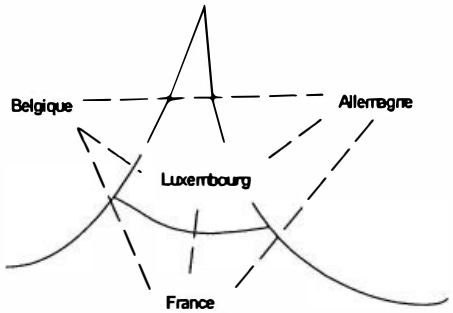


Fig. 11.8 Problème des quatre couleurs : effet Luxembourg.

11.4.4 Un problème d'emploi du temps

Des étudiants doivent s'inscrire à divers cours facultatifs de un jour, à raison de k cours chacun au moins. Ces cours peuvent avoir lieu simultanément, s'ils sont compatibles, i.e. s'ils ne partagent *ni* étudiant(e)(s) *ni* enseignant(e)(s).

Combien faut-il de salles différentes pour faire ces cours dans un minimum de jours?

Construisons un graphe d'incompatibilité, sur la base d'un sommet par cours, et d'une arête entre 2 d'entre eux dès qu'ils ont en commun une personne (étudiant ou enseignant).

En coloriant ce graphe au mieux,

- le nombre de couleurs est le nombre de jours qu'il faut prévoir,
- le nombre de sommets d'une même couleur est le nombre de salles nécessaires aux cours simultanés de la période.

Dans le cas simplifié ci-dessous (tab. 11.1 et fig. 11.9), y oblige à attribuer à B, C, D 3 couleurs différentes. Soit B/0, C/1, D/2. De même v oblige à attribuer à C, D, F trois couleurs différentes ; donc F/0. E doit être ni 0 (u) ni 2(z) ; donc E/1. A ne doit être ni 0 (x) ni 1(t). Donc A/2.

Tableau 11.1 Table des conflits cours/personnes.

Personnes	Cours					
	A	B	C	D	E	F
x	x	x				
y		x	x	x		
z				x	x	
t	x				x	
u		x			x	
v			x	x		x

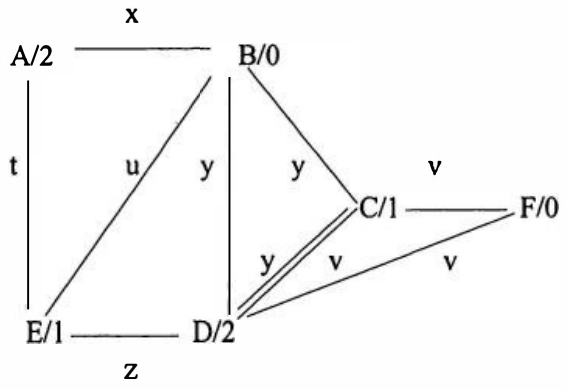


Fig. 11.9 Graphe de conflit cours/personne.

Cet enseignement peut donc s'organiser sur 3 jours différents avec 2 salles, B étant enseigné en parallèle avec F, C en parallèle avec E, A en parallèle avec D.

Tableau 11.2 Organisation possible.

Cours	jours		
salles	J0	J1	J2
S0	B	C	A
S1	F	E	D

11.4.5 Diséquations

DÉFINITION 11.8 – On appelle ***système de diséquations*** des systèmes d'inéquations n'utilisant que l'inégalité.

Considérons le système de diséquations :

$a \neq b$	(1)	$b \neq c$	(2)	$c \neq d$	(3)
$d \neq e$	(4)	$a \neq e$	(5)	$b \neq e$	(6)

Un tel système possède un nombre infini de solutions, dérivables de solutions de base mettant en œuvre un nombre réduit de valeurs.

Associons à ce système un graphe d'incompatibilité G, à raison
- d'un sommet par variable ou constante,
- d'une arête par incompatibilité.

Toute coloration de G constituera une solution de base. Le nombre minimal de valeurs à mettre en œuvre sera précisément le nombre chromatique $\gamma(G)$ du graphe d'incompatibilité. Les solutions finales s'obtiendront en assignant un entier distinct à chaque numéro-de-couleur, chaque constante devant être assignée à un numéro-de-couleur distinct.

Pour le système (1-6) considéré, le graphe associé comporte un triangle (a, b, e) et toute solution est au moins tricolore[6], avec par exemple a = 0, b = d = 1, c = e = 2 (et 5 solutions similaires par permutation des valeurs).

1 sommet/ variable ou constante

1 arête / diséquation

solution :

a : c0 ; b, d : c1 ; c, e : c2.

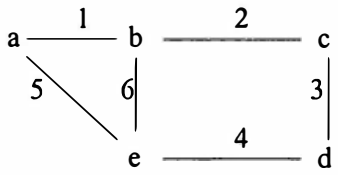


Fig. 11.10 Graphe du système de diséquations.

Ce résultat peut être réemployé dans d'autres contextes, 0, 1 et 2 étant d'abord remplacés par des numéros de couleurs c0, c1, c2 mutuellement exclusifs.

Ainsi, l'ajout de conditions (7) $a \neq 0$ et (8) $d \neq 1$ demanderait seulement que $c0 \neq 0$ et $c1 \neq 1$; soit (c0 c1 c2) valant (1 0 2) ou (1 2 0) ou (2 0 1).

[6] Par la proposition 11.17, le triangle étant assimilé à une clique K_3 ; K_4 demanderait 4 couleurs mais correspondrait à un quadrilatère *complet* ou « tétraèdre ».

11.5 GRAPHES BIPARTIS

Souvent employés en informatique et/ou automatique ils furent étudiés notamment par J. Kuntzmann sous le nom de *réseaux*.

11.5.1 Grafcets

Les grafcets sont utilisés pour spécifier des automatismes, au-delà des automates à états finis (chap. 16).

Les grafcets, ou GRAphe Fonctionnels de Commande Etape/Transition, ont été développés et définis en 1975-77 par le groupe de travail Systèmes Logiques de l'AFCET, puis repris et soutenus par l'Association pour le DEveloppement de la Production Automatisée, normalisés en France et par la Commission Electrotechnique Internationale. Ils sont connus sous d'autres cieux comme *Sequential Functional Chart*.

DÉFINITION 11.9 – Les ***grafcets*** sont des graphes orientés bipartis articulant deux types de sommets :

- les ***étapes***, qui ont un état, actif ou non,
- les ***transitions,*** régissant les évolutions interétapes, et soumises chacune à une condition propre.

CLAUSE DE BIPARTISME. Tout passage d'une étape à une autre se fait par une transition, et réciproquement une étape sépare toujours deux transitions successives[7].

DYNAMIQUE. Un *grafcet* modélise en général un processus de commande d'un système physique.

A une étape est en général liée une action sur ce système, ou une temporisation.

Une transition sera *validée* si *toutes* les étapes qui la précèdent sont actives. La transition sera *franchissable* si sa condition propre ou réceptivité est vraie. Cette réceptivité dépendra des entrées et/ou aux états internes et/ou du temps. La transition devenue validée et franchissable se déclenche, désactivant les étapes précédentes et activant les suivantes.

Une étape peut valider plusieurs transitions, qui peuvent déclencher simultanément, lançant plusieurs *processus concurrents*.

D'autre part, la règle « une transition est *validée* si *toutes* les étapes qui la précèdent sont actives » permet de synchroniser la fin de tels processus concurrents.

DÉFINITION 11.10 – L'activité ou situation d'un grafcet à k étapes est définie par un ***vecteur d'état***, k-uple booléen formé à raison d'une composante par étape.

Un vecteur d'état initial fournit la liste des étapes « initiales », seules actives au départ.

EXEMPLE. Nous empruntons le cas ci-après à une présentation du Grafcet disponible sur Internet et due à Emmanuel Geveaux (fig. 11.11).

Suivant la norme, les *étapes* sont représentées par des carrés, doubles pour les étapes initiales ; à leur droite, un rectangle adjacent précise une action et/ou un commentaire est noté.

Immédiatement avant ou après une *étape*, une barre horizontale dénote :

---

[7] La vérification de cette condition sera un « critère de bonne forme » pour la validation automatique d'un grafcet saisi, par exemple, à l'aide d'un éditeur symbolique.

- une confluence ou convergence OU si elle précède une étape : l'étape devient active si l'une quelconque des transitions regroupées déclenche ;
- une distribution ou divergence si elle suit une étape : l'étape valide – en ce qui la concerne – toute transition qui lui succède.

Les *transitions* sont figurées par les barres horizontales pleines. Le libellé associé spécifie une condition.

Immédiatement avant ou près une *transition*, une double barre horizontale dénote :

- une conjonction ou convergence ET si elle précède une transition : la transition n'est validée que si *toutes* les étapes la précédant sont actives ;
- une distribution ou divergence si elle suit une transition : la transition active simultanément chacune des étapes suivantes.

Le cas proposé est un modèle simplifié pour deux postes couplés : le premier poste usine une pièce et la dépose, le second la prend et l'assemble. Les deux postes ont des marche/arrêt distincts, ils peuvent fonctionner simultanément mais doivent se synchroniser, car il ne peut y avoir qu'une pièce déposée en attente.

Considérons les étapes 12, 1, 13.

L'étape 12 est une attente de dépose, l'étape 1, éventuellement initiale, correspond à une « dépose autorisée ».

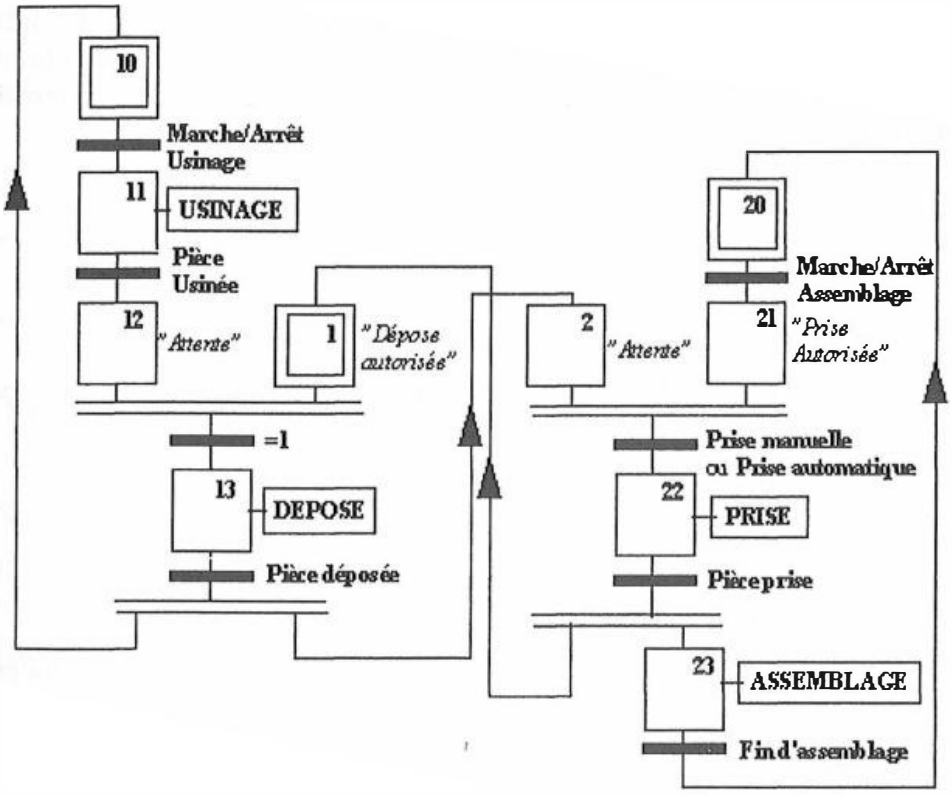


Fig. 11.11 Exemple de grafcet, d'après E.Geveaux.

La double barre horizontale précise que la transition suivante exige l'activité simultanée de 12 et 1 (dépose attendue et autorisée). Quand cette transition est validée, la transition

déclenche instantanément (condition égale à 1, convention pour « toujours satisfaite ») : les étapes 12 et 1 sont désactivées, mais l'étape 13 est activée, déclenchant l'action DEPOSE du système commandé.

Après l'étape 13, la transition « pièce déposée » sera franchie quand on aura à la fois l'étape 13 active (DEPOSE lancée) et un booléen PièceDéposée à vrai (i.e. quand on saura la dépose terminée, le booléen étant un compte-rendu mis à jour par le système physique). Alors, l'étape 13 sera désactivée, tandis que les étapes 2 et 10 seront simultanément activées, 2 en vue d'un nouvel assemblage, et 10 pour un nouvel usinage...

11.5.2 Réseaux de Petri

Dans les ***réseaux de Petri***, on a de même un graphe biparti dont les sommets de répartissent en transitions et en ***places***. L'activité des places est définie par un *nombre de jetons* associé, nul pour une étape inactive. Ainsi,

- l'état d'un réseau de Petri à k places est défini par un vecteur ou k-uple d'entiers positifs,
- les conditions de transitions peuvent porter sur la présence d'un certain nombre de jetons dans les places amont.

Dans les ***réseaux de Petri colorés***, l'activité d'une place est caractérisée par le nombre de jetons qu'elle contient, *regroupés par couleur* [DAVID].

Les réseaux de Petri forment un cadre plus vaste que les grafcets, mais ce cadre des RdP est pratiquement toujours spécialisé pour plus de pertinence : il s'agit plus d'une famille de formalismes que d'un formalisme. Sous réserve de conditions de transitions un peu différentes, les grafcets correspondent grosso modo aux réseaux de Petri dits *saufs*.

11.6 GRAPHES ET HYPERGRAPHES

11.6.1 Définition des hypergraphes

DÉFINITION 11.11– (C. BERGE) – On appelle ***hypergraphe*** H un couple H = <X, U> formé

- d'un ensemble de sommets X,
- d'un ensemble $U \subseteq \mathcal{P}(X)$ d'***hyperarêtes***.

NOTES

- Une hyperarête est un sous-ensemble de X.
- Si toutes les hyperarêtes ont 2 sommets, on est ramené au cas antérieur des graphes symétriques.
- Si tout sommet d'un hypergraphe appartient à au moins une hyperarête, alors la collection des hyperarêtes couvre l'ensemble des sommets.

11.6.2 Représentation matricielle

On représente généralement les hypergraphes par une matrice booléenne rectangulaire, formée à raison d'une ligne par sommet et d'une colonne par hyperarête.

11.6.3 Dualité

DÉFINITION 11.12 – La transposée d'une matrice booléenne rectangulaire M associée à un hypergraphe H est une matrice booléenne rectangulaire M^T représentant un hypergraphe H*, dit hypergraphe dual de H.

Les hyperarêtes de H jouent le rôle de sommets de H*, et les sommets de H le rôle d'hyperarêtes de H*.

PROPOSITION 11.20 (C. BERGE) – *La dualité d'hypergraphes est involutive*, et H** = H.

PREUVE. Si à l'hypergraphe H est associé bijectivement la matrice M, à H* est associé M^T et à H**, M^{TT}. Or la transposition de matrice est involutive : $M^{TT} = M$, donc H** = H. ■

NOTE. Parfois, on trouvera la solution S d'un problème P sur un hypergraphe H en partant de la solution S* du problème dual P* relatif à H*.

11.6.4 Correspondance avec les graphes bipartis

PROPOSITION 11.21 (J. KUNTZMANN) – *A tout hypergraphe* H = <X, U> *correspond un graphe biparti* G = <{X, U} ; W> *où* W = {(x u)| u ∈ U, x ∈ u}.

EXEMPLE. A l'hypergraphe H = <{a, b, c, d}, {u = (a b c), v = (c d)}> correspond le graphe biparti G = <((a b c d) (u v)) ; ((a u)(b u)(c u) (c v)(d v))>.

NOTE. Ce théorème autorise un « raisonnement spiral » graphe simple → hypergraphe → graphe biparti... ; mais si H a N sommets et M hyperarêtes, G a M+N sommets *plus* de l'ordre de M à M·N arêtes.

11.6.5 Correspondance avec les multigraphes

PROPOSITION 11.22 – A tout hypergraphe H = <X, U> correspond un multigraphe G = <X, K > où K est une collection de cliques de X, sur la base d'une clique dans K par hyperarête u ∈ U de H.

11.6.6 Hypergraphes et systèmes d'équations

Soit un système d'équations :

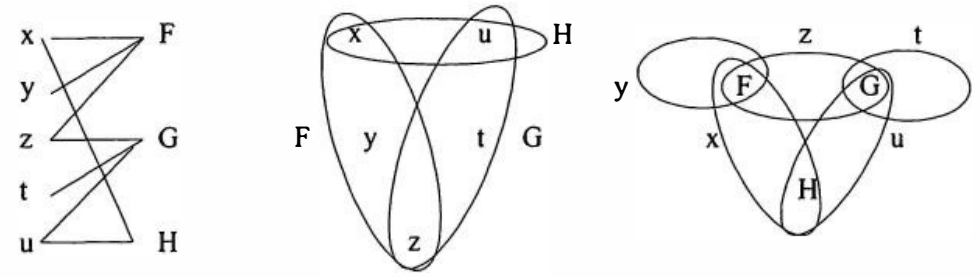
F(x, y, z) = 0 ;
G(z, t, u) = 0 ;
H(x, u) = 0.

On peut représenter ce système d'interdépendances par

- la table 11.3 ci-contre,
- les (hyper)graphes de la figure 11.12.

Tableau 11.3 Dépendances.

fonctions / variables	F	G	H
x	x		x
y	x		
z	x	x	
t		x	
u		x	x



graphe biparti

hypergraphe
1 sommet/ variable
1 hyperarête / équation
« F lie x, y, z »

hypergraphe dual
1 sommet/ équation
1 hyperarête / variable
« x lie F et H »

Fig. 11.9 Dépendances : graphe biparti, hypergraphe, hypergraphe dual.

Pour les systèmes complexes d'équations, aux considérations classiques sur les degrés de liberté, s'ajouteront des considérations topologiques sur la ***structure de dépendance***. Soit un système physique modélisé par un système d'équations auquel on attache un graphe de dépendance. La classification ***variables exogènes / endogènes***, mais aussi la détection des sommets du graphe de dépendance formant un *ensemble d'articulations* guideront la fragmentation d'un gros système en sous-systèmes aux inter-couplages aussi simples et réduits que possible.

EXEMPLE. Dans le projet Ampère 3, le physicien-programmeur est invité à déclarer ses équations algébriques plutôt qu'à les résoudre : le compilateur doit insérer une méthode adéquate lorsqu'on a besoin de solutions, l'équation étant alors invoquée avec certaines variables précisées.

Un élément fondamental des stratégies développées est de considérer, sous conditions, une équation à n variables comme une fonction établissant une variable quand les n – 1 autres sont connues.

Ainsi, $G(z, t, u) = 0$ engendre les 3 interprétations formelles : $z = g_1(t, u)$, $t = g_2(z, u)$, $u = g_3(z, t)$ (J.F. Martin).

Le système précédent lie 5 variables par 3 équations. Supposons t et u connues. Alors G fixe $z = g_1(t, u)$, et H fixe $x = h_1(u)$. Par suite, F fixe $y = f_2(x, z) = f_2(h_1(u), g_1(t, u))$. Si ce n'est pas la solution, c'est au moins un plan d'action...

11.6.7 Hypergraphes et bases de données

Considérons une base de données regroupant des relations f, g, h... partageant certaines de leurs dimensions A, B, C, D... ; il pourra être intéressant de la modéliser par un hypergraphe, à raison d'un point par dimension, et d'une hyperarête par relation.

11.6.8 Hypergraphes, multicoloration, classification

Considérons une collection X d'objets ou d'individus entre lesquels on repère des (dis)similarités, et modélisons-la par un hypergraphe, en prenant X comme ensemble de sommets. Regroupons ensuite en une même hyperarête les éléments de X partageant une similarité définie, et qui recevront une même *couleur* interprétable par un caractère commun.

A chaque sommet est maintenant associé un spectre de couleurs, fixé par la collection des hyperarêtes auxquelles il appartient. Ce spectre, concrétisé par un vecteur booléen, à raison d'une composante par hyperarête, permet de définir une distance de Hamming entre éléments de X, donc d'amorcer une classification en termes d'analogies et de différences.

Tableau 11.4 Individus, attributs, distances.

individu	marin	a des ailes	a des pattes	a des roues	a un moteur	distances					
albatros	x	x	x			albatros					
avion		x		x	x	4	avion				
exocet	x	x				1	3	exocet			
mouche		x	x			1	3	2	mouche		
rat			x			2	4	3	1	rat	
automobile				x	x	5	1	5	4	3	auto

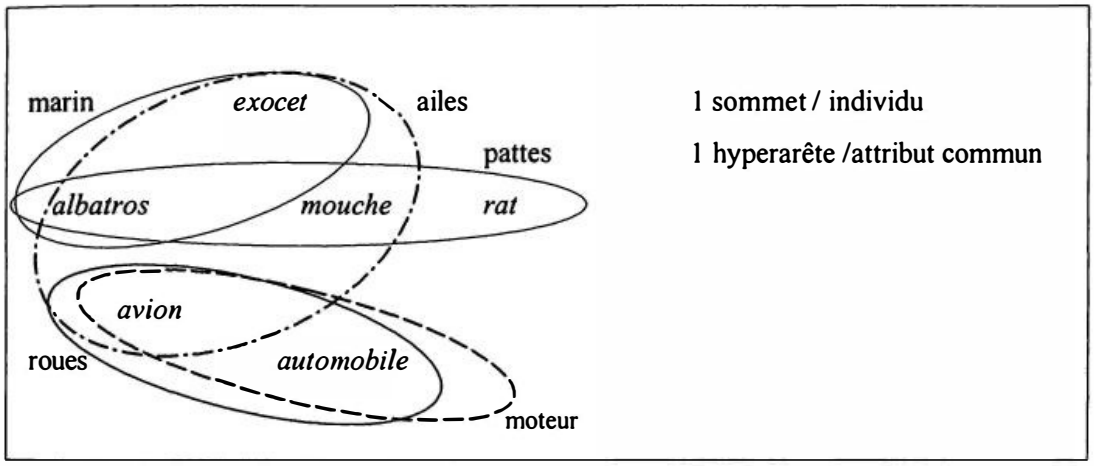


Fig. 11.10 Hypergraphe individus/attributs.

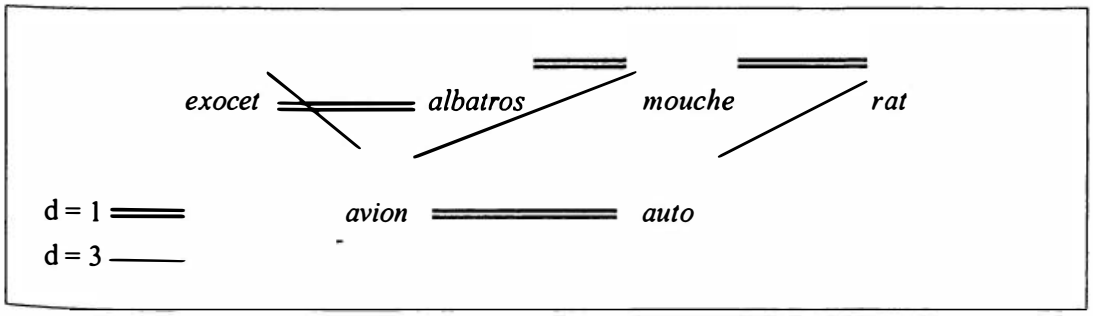


Fig. 11.11 Graphe des distances.

11.7 EXERCICES

11.7.0 Part à deux

2 joueurs sont en face d'un tas d'allumettes.

Chaque joueur peut enlever à son tour une ou plusieurs allumettes, mais au plus la moitié des allumettes restantes (ex : pour 6 ou 7 allumettes, il peut en enlever de 1 à 3).

Celui qui reçoit la dernière allumette a perdu.

Définir une stratégie gagnante si :

- le premier joueur reçoit 15 allumettes,
- le premier joueur reçoit un nombre d'allumettes pris au hasard entre 7 et 100.

11.7.1 Primus inter pares

2 joueurs sont en face d'un tas de jetons. Chaque joueur peut enlever à son tour un *nombre premier* de jetons. Celui qui prend le dernier jeton a gagné.

Définir une stratégie gagnante si :

- le premier joueur reçoit 15 jetons,
- le premier joueur reçoit un nombre de jetons pris au hasard entre 10 et 100,
- le premier joueur reçoit un nombre de jetons pris au hasard au-delà de 100.

11.7.2 Covoiturage

Alfred, Béatrice, Charlotte, Didier, Evelyne et François, collègues de travail, doivent aller à une même réunion à 200 km de là. On ne va pas prendre chacun sa voiture ! Mais Alfred ne supporte pas Charlotte, qui ne supporte pas Evelyne, qui ne supporte pas Béatrice, qui ne supporte pas Didier, qui ne supporte pas François, qui ne supporte pas Evelyne.

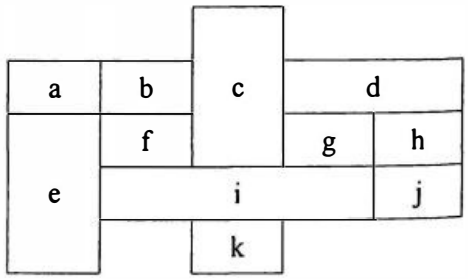
Composer un nombre minimal de voitures pour que le voyage se passe au mieux.

11.7.3 Graphes équilibrés

On considère des ***sociogrammes***, ou graphes d'affinité, dans lesquels les sommets représentent des personnes, et les relations R^+ et R^- les relations d'affinité /hostilité. Comment trouver une partition de l'ensemble des personnes telle qu'aucune des classes formées ne contienne de couple de R^- ? Discuter des clans ainsi formés en termes de tension interne ? de solidité ? Dans ce contexte, que signifie une clique de R^+ ?

11.7.4 Coloriage de carte

Combien de couleurs sont nécessaires et suffisantes pour colorier la carte ci-contre, de telle façon que 2 zones séparées par un trait (et non pas seulement par un point) soient toujours de couleurs différentes ?



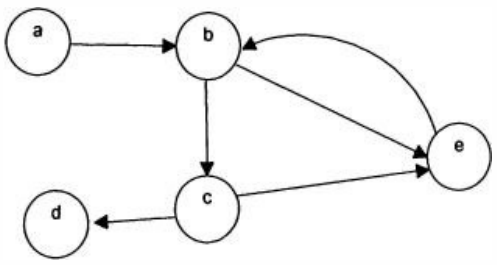
11.7.5 Jeu sur un graphe

Un jeton est placé en *a* du graphe ci-dessous. Chaque joueur le déplace à son tour d'un sommet à l'autre, suivant un des arcs partant du sommet où se trouve le jeton. Celui qui arrive en *d* a terminé.

Donner la liste des mouvements gagnants

- si celui qui arrive en d a gagné,
- si celui qui arrive en d a perdu.

On pourra s'aider du calcul de fonction(s) de Grundy.



11.7.6 Coloriage d'un cube

On désire *colorier un cube*, sans utiliser la même couleur pour les éléments adjacents. Combien faut-il de couleurs pour colorier

- les sommets ?
- les arêtes ?
- les faces ?

11.7.7 Mise en page

Les metteurs en page du *Libertaire* ont un principe : pour un espace sans frontière, ne jamais séparer deux articles par un filet (ou trait). Pour éviter la confusion entre un article et son voisin, il faudrait jouer sans abuser sur les polices et leurs attributs. En alternant romain et italique, combien de polices suffiront à toujours contraster les textes adjacents ?

11.8 PROBLÈMES

11.8.1 Itinéraires urbains

Considérons une ville, ses rues et leurs intersections. Une telle ville est découpée en quartiers par des « obstacles naturels » tels que rivières, lignes de chemins de fer ; cependant, les quartiers sont reliés par des points de franchissement ou d'autres liaisons (boulevard, autoroute urbaine...).

Au-delà de quelques centaines d'intersections, le graphe bâti sur les seules intersections et segments de rues devient peu lisible, et on est amené à distinguer un niveau local, pour les chemins internes à un quartier, et un niveau global, interquartier.

Montrer que les liaisons inter-quartiers peuvent être modélisées par un multigraphe, à raison d'un sommet par quartier et d'un arc ou arête par point de franchissement ou voie de liaison inter-quartier (boulevard, autoroute...).

Montrer que ces liaisons inter-quartiers peuvent aussi être modélisées par un hypergraphe, à raison d'un sommet par liaison ou point de franchissement, et d'une hyperarête par quartier.

11.8.2 Jeu sur un graphe

Un jeton est placé en « a » du graphe du problème 11.7.5. Chaque joueur le déplace à son tour d'un sommet à l'autre, suivant un des arcs partant du sommet où se trouve le jeton. Celui qui arrive en « d » a terminé.

Donner la liste des mouvements gagnants si celui qui arrive en d a gagné à condition d'être allé *un nombre impair de fois* en e.

11.8.3 Les cartes postales

Vous êtes en vacances dans cette station réputée, mais il pleut. C'est le moment d'envoyer des cartes postales à une vingtaine de parents et amis ! Malheureusement, il n'existe que k types distincts de cartes. Cependant, vous savez bien qui connaît qui.

Comment donner l'impression à chacun de recevoir une carte postale originale ?
Est-ce encore possible si les personnes concernées forment des cliques ?
Jusqu'où pouvez-vous tenir compte des goûts de vos correspondants ?

11.9 PISTES DE RÉFLEXION

11.9.0 Emploi du temps

Reformuler le paragraphe 11.4.4 en termes d'hypergraphes.

11.9.1 Fonction de Grundy, destin et libre-arbitre

On suppose la vie d'une personne modélisée[8] par un graphe, à raison d'un sommet par situation (personne, environnement) et d'un arc par action, ces actions étant soit une action de la personne sur son environnement soit une (ré)action de l'environnement sur la personne. On suppose de plus que ce graphe possède une fonction de Grundy.

Montrer que dans ces conditions, la personne peut être soumise à un destin inexorable malgré des choix très ouverts à chaque instant. A quelles conditions son libre-arbitre peut-il s'exercer fructueusement ?

Que vous acceptiez ou non ce modèle, quels sont, à votre avis, les points d'accord et les divorces entre la réalité et ce modèle ? Essayer de pondérer leur importance relative avant de conclure.

11.10 LECTURES

BERGE C., *Théorie des graphes et ses applications,* CUM n° 2, Dunod, 1963.
Graphes, Gauthier-Villars, 1983.
Hypergraphes (combinatoire des ensembles finis), Gauthier-Villars, 1987.

DAVID R. et ALLA H., *Du Grafcet aux réseaux de Petri,* collection Traité des Nouvelles Technologies, série Automatique, Hermès, 1989.

GEVEAUX E., *Introduction au Grafcet*, sur www.tecatlant.fr

KUNTZMANN Jean, *Théorie des réseaux,* Dunod, 1972.

NUSSBAUMER H., *Informatique industrielle,* vol. 2, chap. 5, Presses polytechniques et universitaires romandes, 1986.

Pour la Science n° 2, *La solution du problème des 4 couleurs*, déc. 1977, pp. 56-70.

coll., *Petite encyclopédie des mathématiques*, chap. XXXVI, Edition Pagoulatos.

[8] Tout modèle est une hypothèse, à valider sur le réalisme de ses conséquences.

TROISIÈME PARTIE

ALGÈBRE

CHAPITRE 12

OPÉRATEURS & ALGÈBRES

Ce chapitre passe en revue les propriétés les plus classiques des opérateurs et les éléments remarquables qui peuvent leur être attachés. Il dérive d'une longue histoire, qui commence avec l'apparition des signes + et – dans des livres allemands des années 1480, prémices de la notation algébrique actuelle.

Ce chapitre sera notamment utile :

- en *compilation* et en *transformation de programmes* (optimisation, parallélisation),
- en *calcul formel,*
- pour la programmation dans tous les *langages à opérateurs,* tels que APL, Algol 68, Prolog, Ada 83/95, C^{++}, Java et, à un moindre degré, LISP et les occams 1/2/3,
- pour la *recherche de méthodes et algorithmes génériques.*

12.1 OPÉRATEURS & SIGNATURES

12.1.1 Opérateur

DÉFINITION 12.1 – Un ***opérateur*** o dénote une application $o : A \otimes B \otimes C \rightarrow D$, ou à défaut une fonction.

12.1.2 Arité d'un opérateur

DÉFINITION 12.2 – L'arité d'un opérateur est le nombre d'***arguments*** ou ***opérandes*** qu'admet cet opérateur (cf. § 3.10.1).

Un ***opérateur unaire*** (ou 1-aire ou d'arité 1) admet un opérande.
ex : succ : $\mathbb{N} \rightarrow \mathbb{N}$.

Un ***opérateur binaire*** (ou 2-aire ou d'arité 2) demande deux opérandes…
ex : + : $\mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$.

Un opérateur $o : A \otimes B \otimes C \rightarrow D$ est un ***opérateur ternaire*** (ou 3-aire ou d'arité 3).

12.1.3 Signature et profil d'un opérateur

DÉFINITION 12.3 – Un opérateur $o : A \otimes B \otimes C \rightarrow D$ est dit de ***signature*** $A \otimes B \otimes C$ et de ***profil*** $A \otimes B \otimes C \rightarrow D$

On appelle ***opérateur interne*** à A, d'arité n, une application de *profil* $A^n \rightarrow A$. C'est le cas des additions et multiplications numériques, des additions de vecteurs et du produit vectoriel.

On appelle ***opérateur externe*** une application de *profil* $A^n \rightarrow B$. C'est le cas des comparateurs ou du produit scalaire.

On appelle ***opérateur hétérogène*** une application de *profil* $A \otimes B \otimes C \ldots \otimes K \rightarrow Z$. C'est le cas des sélecteurs, du réplicateur de chaîne…

12.1.4 Stabilité d'un opérateur interne

DÉFINITION 12.4 – Soit un opérateur $\tau : A^n \rightarrow A$ *interne* à A, et $B \subseteq A$:

τ sera dit ***stable pour*** B ssi $\tau : B^n \rightarrow B$

EXEMPLE. Soit $\mathbb{N}$, l'ensemble des entiers naturels, muni de l'addition $+ : \mathbb{N} \otimes \mathbb{N} \rightarrow \mathbb{N}$.

- *Le sous-ensemble des entiers pairs est stable,* car pair + pair → pair,
- Le sous-ensemble des entiers impairs ne l'est pas, car impair + impair → pair.

12.2 SYNTAXE DES EXPRESSIONS

12.2.1 Opérateurs unaires

Les ***opérateurs unaires*** s'appliquent à un seul argument ; ils sont généralement dénotés

- par une ***notation préfixée*** dans laquelle

  f x signifie f(x)

  Ex : + et – unaires, sin, cos … √
- ou plus rarement ***postfixée*** avec : *x f pour f(x)* comme dans n!

12.2.2 Opérateurs binaires

Les ***opérateurs binaires*** s'appliquent à un couple d'arguments ; ils sont généralement désignés par une ***notation infixée***, dans laquelle

x f y signifie f(x,y)

Ex : +, –, ×, /, …

12.2.3 Opérateurs d'arités supérieures

Il n'y a pas de règle générale pour les arités supérieures, la notation préfixée tendant à se confondre avec un appel de fonction.

On admet cependant des écritures infixes (smalltalk…).

EXEMPLE. Calcul d'un signe par utilisation infixe de l'opérateur ternaire de choix *cond ? valSiVrai : valSiFaux* en C.

signe(x) ≡ (x > 0 ? '+' : x = 0 ? '0' : '–')

12.2.4 Expressions

Chaque opération dénote une valeur, et peut donc être employée comme argument d'une écriture plus complexe.

On dit une ***expression syntaxiquement bien formée*** dès lors que les arités des opérateurs (resp. des fonctions) sont respectées. Elle est non ambiguë si l'association de chaque opérateur avec son ou ses opérandes est univoque.

EXEMPLE. Une expression telle que x♣y♦z♥t est syntaxiquement correcte si les opérateurs sont d'arité 2, mais ambiguë : ♦ combine quoi à quoi ?

La définition suivante élimine toute ambiguïté des expressions.

DÉFINITION 12.5 – Une ***expression*** dénote soit un *opérande*, soit un *opérateur n-aire et ses n opérandes,* soit une *valeur de fonction*. Un ***opérande*** est soit une *constante,* soit une *variable,* soit une *expression entre parenthèses.*

Pour utiliser simultanément sans ambiguïté des opérateurs pré- et postfixés, il faudrait parenthèser même les expressions réduites à un opérateur unaire et son opérande, afin de bien distinguer

- n + (4!), somme de n et de la factorielle de 4,
- (n + 4)!, factorielle de la somme de n et 4.

12.3 SÉMANTIQUE FORMELLE DES EXPRESSIONS

Vue de l'extérieur, une expression entre parenthèses est un simple opérande, même si sa structure interne est complexe. La structure interne, qui guide l'évaluation, est à base de termes, constantes et variables, combinés par des opérations ou des appels de fonctions produisant de nouveaux termes.

DÉFINITION 12.6 – On dira une ***expression sémantiquement bien formée,*** *à valeur dans A,* si elle dénote

- soit un opérande à valeur dans A,
- soit un opérateur n-aire de profil $A_1 \otimes A_2 \ldots \otimes A_n \rightarrow A$ et ses n opérandes, chaque opérande étant compatible, rang par rang avec la signature (le premier opérande à valeur dans A_1, le second à valeur dans A_2, ... le dernier à valeur dans A_n), soit une valeur de fonction.

Un ***opérande*** à valeur dans A est soit une constante de A, soit une variable à valeur dans A, soit une expression entre parenthèses à valeur dans A.

NOTE. Deux opérations différentes peuvent être dénotées sans ambiguïté par le même symbole *ssi* leurs *signatures* sont *disjointes* : c'est la clé de la technique de la ***surcharge*** (Algol 68, Ada 83/95, C^{++}...).

12.4 ALGÈBRES

Cette section pourra être sautée en première lecture.

12.4.1 Algèbres formelles

DÉFINITION 12.7 – On appelle ***algèbres formelles*** des structures $<\Sigma ; \Phi>$ où

- Σ désigne une collection d'ensembles formels ou ***sortes***,
- $\Phi = \{\Phi_0, \Phi_1, \Phi_2, \ldots \Phi_n\}$ est une collection de Φ_k, chacun des Φ_k étant collection de symboles fonctionnels k-aires, de profils relatifs à tout ou partie des sortes de Σ.

Cette définition est suffisante si on assimile :

- les constantes remarquables à des fonctions 0-aires,
- les opérateurs à des fonctions,
- les relations $R \subseteq A \otimes B \otimes C$ à des applications $R : A \otimes B \otimes C \rightarrow \mathbb{B}$.

12.4.2 Algèbres de termes

DÉFINITION 12.8 – En se donnant pour chaque sorte un jeu de variables, une algèbre formelle engendre, de façon purement syntaxique, l'***algèbre des termes licites*** (i.e. correctement écrits).

EXEMPLE. Soit l'algèbre formelle $<S ; 0, s : S \rightarrow S>$

- on peut écrire les termes clos s(0), s(s(0)), s(s(s(0))) ...
- si x est une variable à valeur dans S, on peut écrire de même des termes s(x), s(s(x))...

12.4.3 Interprétations

DÉFINITION 12.9 – On appelle ***interprétation*** $<S, F>$ ***d'une algèbre formelle*** $<\Sigma, \Phi>$ tout système obtenu par une substitution cohérente.

EXEMPLE. $<S ; 0, s : S \rightarrow S>$ admet comme interprétations

- $<\mathbb{N} ; 0 ; s(n) = n + 1>$, et les 0, s(0), s(s(0)), s(s(s(0)))... correspondent alors respectivement à 0, 1, 2, 3, ... *valeurs distinctes les unes des autres* (Peano),
- $<\mathbb{C} ; 1 + 0i, s(z) = ze^{i\vartheta}>$. Les 0, s(0), s(s(0)), s(s(s(0)))... correspondent respectivement au point 1 + 0i et à ses images par des rotations successives autour de l'origine, d'angle ϑ ; *la suite des s^i étant périodique ssi ϑ exprime un nombre rationnel de tours.*

12.4.4 Modèles

Considérons une algèbre formelle $<\Sigma, \Phi>$ muni d'une théorie, i.e.

- un ensemble initial A_Σ de propriétés nécessaires et suffisantes, dites ***axiomes,*** formant le germe de la théorie de Σ,
- un ensemble Π de propriétés dérivées.

Ce qu'on note : $<\Sigma, \Phi> \models A_\Sigma \Rightarrow \Pi$ pour signifier que, pour la structure $<\Sigma, \Phi>$, la satisfaction de A_Σ entraîne celle de Π.

Soit maintenant $<S, F> = \text{int}(<\Sigma, \Phi>)$ une interprétation de $<\Sigma, \Phi>$.

Si l'interprétation A_S des axiomes A_Σ est vraie dans $<S, F>$, *alors*

- $<\Sigma, \Phi>$ est dit ***modèle*** pour $<S, F>$,
- l'interprétation P des propriétés Π est vraie dans $<S, F>$: la théorie élaborée pour le modèle s'applique à l'interprétation dès lors qu'elle satisfait aux axiomes.

Cette situation pourra se noter :

$$\text{int} : (<\Sigma,\Phi> \models A_\Sigma \Rightarrow \Pi) \rightarrow (<S, F> \models A_S \Rightarrow P)$$

12.4.5 Catégories

DÉFINITION 12.10 – Une ***catégorie*** sera la collection d'un modèle $<\Sigma, \Phi> \models A_\Sigma \Rightarrow \Pi$, et de toutes les interprétations qui satisfont ses axiomes. Ses propriétés sont à l'origine des méthodes et algorithmes génériques.

Ainsi, la catégorie des monoïdes, notée *Monoïde*, sera formée

- du système formel archétype ou modèle de la catégorie, auquel sont attachées une axiomatique et des propriétés génériques,

- de la classe[1] Obj(Monoïde) de toutes les structures qui (à l'interprétation près) en respecteront les axiomes, et, *ipso facto*, bénéficieront (de l'interprétation) des propriétés et méthodes génériques.

Une catégorie peut être dérivée d'une autre par spécialisation, d'où des hiérarchies de catégories (ABRIAL, Colloque GROPLAN, Cargèse, 1979).

Une catégorie peut également être obtenue par combinaison de plusieurs autres.

La spécification d'un modèle ou d'une catégorie sera souvent basée sur les propriétés des opérateurs, dont nous examinons maintenant les plus classiques.

NOTE. Les chapitres 13 à 15 peuvent être considérés comme présentant les catégories Monoïde (et Groupe), Dioïde (et Anneau) et Algèbre de Boole.

12.5 PROPRIÉTÉS DES OPÉRATEURS

12.5.1 Involution

DÉFINITION 12.11 – Un *opérateur unaire* $\blacklozenge$ interne à un ensemble X sera dit ***opérateur involutif*** ssi $\forall\, x \in X \;\; \blacklozenge\blacklozenge x = x$.

EXEMPLES : moins unaire, négation logique, symétries par rapport à un point ou un axe, inversion, transposition de matrices...

12.5.2 Associativité

DÉFINITION 12.12 – Un *opérateur binaire* $\blacklozenge$ interne à un ensemble X sera dit ***opérateur associatif*** ssi

$$\forall x,y,z \in X \quad ((x\blacklozenge y)\blacklozenge z) = (x\blacklozenge(y\blacklozenge z))$$

On convient de noter $x\blacklozenge y\blacklozenge z$ cette valeur commune.

EXEMPLES : +, ×, max, min, pgcd, ppcm, et, ou, ouExclusif, union, intersection, différenceSymétrique *mais pas* –, / , ni, ...

PROPOSITION 12.1 – L'associativité rend le résultat de la combinaison de n éléments indépendant de l'ordre de groupement ou parenthésage des termes. Ainsi :

$$((x\blacklozenge y)\blacklozenge z)\blacklozenge t = (x\blacklozenge(y\blacklozenge z))\blacklozenge t = x\blacklozenge((y\blacklozenge z)\blacklozenge t) = x\blacklozenge(y\blacklozenge(z\blacklozenge t)), \text{ noté } x\blacklozenge y\blacklozenge z\blacklozenge t\ldots$$

12.5.3 Commutativité

DÉFINITION 12.13 – Un *opérateur binaire* $\blacklozenge$ interne à X sera dit ***opérateur commutatif*** ssi

$$\forall x,y \in X \quad (x\blacklozenge y) = (y\blacklozenge x)$$

EXEMPLES : +, ×, max, min, pgcd, ppcm, et, ou, ouExclusif, ni, union...
mais pas –, / ...

[1] Pour éviter certains paradoxes, on appelle classe toute collection de classes ou d'ensembles, en réservant le nom d'ensemble aux collections d'éléments. On parlera de la classe universelle, formée de tous les ensembles ; de la classe des ensembles satisfaisant une propriété P, et de la classe « complémentaire » des ensembles qui ne la satisfont pas...

PROPOSITION 12.2 – Pour un opérateur ♦ associatif et commutatif, le résultat de la combinaison de n éléments est indépendant de l'ordre des termes.

Ainsi, $(x \blacklozenge y) \blacklozenge (z \blacklozenge t) = x \blacklozenge y \blacklozenge z \blacklozenge t = t \blacklozenge x \blacklozenge y \blacklozenge z = (t \blacklozenge ((x \blacklozenge y) \blacklozenge z)$.

12.5.4 Idempotence

DÉFINITION 12.14 – Un *opérateur binaire* ♦ interne à X sera dit ***opérateur idempotent*** ssi

$$\forall x \in X \; x \blacklozenge x = x$$

EXEMPLES : max, min, pgcd, ppcm, et, ou, union, intersection…
mais pas +, – ,×, / , ni …

12.5.5 Simplifiabilité

DÉFINITION 12.15 – Un ***opérateur binaire*** ♦ sera dit :

- ***simplifiable à gauche*** ssi $\forall x, y, z \quad x \blacklozenge y = x \blacklozenge z \Rightarrow y = z$
- ***simplifiable à droite*** ssi $\forall x, y, z \quad x \blacklozenge y = z \blacklozenge y \Rightarrow x = z$
- ***(bi-)simplifiable*** ssi *simplifiable à gauche ET à droite.*

EXEMPLES. + pour ℝ, × pour ℝ– { 0 },

$x + 3 = 4 + 3$	entraîne bien $x = 4$,
$z \times 3 = 4 \times 3$	entraîne bien $z = 4$,
$3 \times z = 3 \times 4$	entraîne bien $z = 4$,
mais $0 \times 3 = 0 = 0 \times 7$	*n'entraîne pas* $3 = 7$.

NOTE. Cette propriété, autorisant des simplifications membre à membre, facilite la *résolution des équations utilisant l'opérateur* ♦.

12.5.6 Caractérisation d'un opérateur

EXEMPLE. Caractériser dans ℝ l'opérateur : $x \blacklozenge y \equiv (x+y)/2$.

- Il est *commutatif* car on a toujours $x \blacklozenge y = (x+y)/2 = (y+x)/2 = y \blacklozenge x$.
- Il est *idempotent* car on a toujours $x \blacklozenge x = (x+x)/2 = x$.
- Il *n'est pas associatif* car $((x \blacklozenge y) \blacklozenge z) = ((x+y)/2) \blacklozenge z = (x+y)/4+z/2$
  mais $(x \blacklozenge (y \blacklozenge z)) = (x \blacklozenge ((y+z)/2)) = x/2+(y+z)/4$.
- Il est *simplifiable* car on a toujours
  $x \blacklozenge y = x \blacklozenge z \Rightarrow (x + y)/2 = (x + z)/2 \Rightarrow x + y = x + z \Rightarrow y = z$,
  et, par commutativité, $x \blacklozenge y = z \blacklozenge y \Rightarrow y \blacklozenge x = y \blacklozenge z \Rightarrow x = z$.

NOTE. Les propriétés examinées permettent de décider de l'équivalence de formules :

- l'idempotence, l'involution et la régularité autorisent des *simplifications,*
- associativité et commutativité des *transformations.*

12.6 ÉLÉMENTS REMARQUABLES POUR UN OPÉRATEUR

12.6.1 Point(s) fixe(s)

DÉFINITION 12.16 – Dans un ensemble E, on appelle ***point fixe d'un opérateur unaire*** ♠ un élément f de E tel que

$$f = \spadesuit f$$

EXEMPLES

- Dans $\mathbb{R}$, le *moins unaire* et *sin* ont zéro comme point fixe.
- Dans $\mathbb{R}^+$, $\sqrt{}$ admet 0 et 1 comme points fixes.
- Mais dans $\mathbb{N}$, succ : $n \to n+1$ n'a pas de point fixe.

12.6.2 Eléments neutres

DÉFINITION 12.17 – Pour un opérateur binaire $\blacklozenge$ interne à X, l'***élément*** e de X est

- ***neutre à gauche*** ssi $\forall x \in X \quad e \blacklozenge x = x$
- ***neutre à droite*** ssi $\forall x \in X \quad x \blacklozenge e = x$
- ***(bi-)neutre*** ssi $\forall x \in X \quad e \blacklozenge x = x = x \blacklozenge e$

PROPOSITIONS

12.3 – Une même opération ne peut avoir un élément neutre à gauche et un élément neutre à droite distincts.

PREUVE. Soient e_1 élément neutre à gauche pour $\blacklozenge$, et e_2 élément neutre à droite *distinct*: alors $e_1 \blacklozenge e_2 = e_1$ car e_2 est neutre à droite, et $e_1 \blacklozenge e_2 = e_2$ car e_1 est neutre à gauche ; donc $e_1 = e_2$, ce qui est contradictoire. D'où la proposition. ■

12.4 – Un élément neutre est unique.

PREUVE. Soient e_1 et e_2 deux éléments neutres *distincts* pour $\blacklozenge$: alors $e_1 \blacklozenge e_2 = e_1$ car e_2 est neutre à droite, et $e_1 \blacklozenge e_2 = e_2$ car e_1 est neutre à gauche ; donc $e_1 = e_2$, ce qui est contradictoire. D'où l'unicité de l'élément neutre. ■

12.6.3 Eléments absorbants

DÉFINITION 12.18 – Pour un opérateur binaire $\blacklozenge$ interne à X, l'***élément*** a de X est

- ***absorbant à gauche*** ssi $\forall x \in X \quad a \blacklozenge x = a$
- ***absorbant à droite*** ssi $\forall x \in X \quad x \blacklozenge a = a$
- ***(bi-)absorbant*** ssi $\forall x \in X \quad a \blacklozenge x = a = x \blacklozenge a$

PROPOSITIONS

12.5 – Une opération ne peut avoir un élément absorbant à gauche et un élément absorbant à droite distincts.

12.6 – Un élément absorbant est unique.

PREUVE. Soient e_1 et e_2 deux éléments absorbants pour $\blacklozenge$: alors $e_1 \blacklozenge e_2 = e_1$ car e_1 est absorbant à gauche, $e_1 \blacklozenge e_2 = e_2$ car e_2 est absorbant à droite ; donc $e_1 = e_2$. Deux éléments absorbants sont donc identiques. ■

12.6.4 Eléments involutifs

DÉFINITION 12.19 – Pour un opérateur binaire $\blacklozenge$ d'élément neutre e, i est un ***élément involutif*** ssi $i \blacklozenge i = e$.

EXEMPLES

- -1 est involutif pour $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ dotés de la multiplication.
- Soit $x \blacklozenge y = (x \cdot y) \bmod (n-1)\cdot(n+1)$ avec $n \geq 2$: *n est involutif pour* $\blacklozenge$ car $(n \cdot n) \bmod (n^2-1) = 1$. Ainsi, 2, 3, 4, 5, … sont les *éléments involutifs* pour la multiplication *modulo* 3, 8, 15, 24, …
- $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ est involutive pour le produit des matrices 3×3

PROPOSITION 12.7 – Tout élément neutre (resp : à gauche, à droite) est involutif.

Ainsi, $0 + 0 = 0$, $1 \cdot 1 = 1$, *mais aussi* $0 - 0 = 0$, $1/1 = 1$, $1^1 = 1$...

12.6.5 Eléments idempotents

DÉFINITION 12.20 – Pour un opérateur binaire ♦, i est un ***élément idempotent*** ssi

$$i \blacklozenge i = i$$

EXEMPLE. Soit $x \blacklozenge y = (x \cdot y) \bmod n \cdot (n-1)$ avec $n \geq 3$: n est idempotent pour ♦ car $(n \cdot n) \bmod (n^2 - n) = n$. Ainsi, 3, 4, 5, 6, 7 ... sont les *idempotents* relatifs aux modules 6, 12, 20, 30, 42...

PROPOSITION 12.8 – Pour une opération idempotente, tout élément est idempotent.

PROPOSITION 12.9 – Un élément neutre (resp : à gauche, à droite) ou absorbant (resp : à gauche, à droite) est idempotent.

Ainsi, $1 \cdot 1 = 1$, $0 - 0 = 0$, $1^1 = 1$ ...

12.6.6 Eléments simplifiables

Ces éléments[2] autorisent des *simplifications membre à membre.*

DÉFINITION 12.21 – Pour un opérateur binaire ♦, l'***élément*** r est

- ***simplifiable à gauche*** ssi $\forall x,y\ r \blacklozenge x = r \blacklozenge y \Rightarrow x = y$
- ***simplifiable à droite*** ssi $\forall x,y\ x \blacklozenge r = y \blacklozenge r \Rightarrow x = y$
- ***(bi-)simplifiable*** ssi simplifiable à droite et simplifiable à gauche.

PROPOSITION 12.10 – Pour un opérateur binaire ♦, si l'***élément*** r est

- ***simplifiable à gauche*** on a $\forall x,y\ \ x \neq y \Rightarrow r \blacklozenge x \neq r \blacklozenge y$,
- ***simplifiable à droite*** on a $\forall x,y\ \ x \neq y \Rightarrow x \blacklozenge r \neq y \blacklozenge r$.

PREUVE. Par contraposition de la définition. ■

PROPOSITION 12.11 – Tout élément neutre à gauche (resp. à droite) est simplifiable à gauche (resp. à droite).

Dans le cas d'ensembles comportant peu d'éléments, l'opérateur binaire peut être défini par une table. Alors, la proposition 12.10 donne :

- un élément est simplifiable à droite si tous les éléments de sa colonne sont distincts,
- un élément est simplifiable à gauche si tous les éléments de sa ligne sont distincts.

Si l'opérateur est commutatif, tout élément simplifiable à gauche l'est à droite et réciproquement. Alors, tout élément simplifiable d'un côté est simplifiable tout court.

EXEMPLES

- Tout nombre est simplifiable pour l'addition (al Kwarizmi, ~820).
- Tout réel non nul est simplifiable pour la multiplication.
- Pour l'exponentiation,

[2] Dits aussi *éléments réguliers.*

- $0\uparrow x = 0\uparrow y$ n'entraîne pas $x = y$ et $x\uparrow 0 = y\uparrow 0$ n'entraîne pas $x = y$: 0 n'est pas simplifiable pour $\uparrow$
- $1\uparrow x = 1\uparrow y$ n'entraîne pas $x = y$, mais $x\uparrow 1 = y\uparrow 1$ entraîne $x = y$ par définition, donc 1 est *simplifiable à droite* pour $\uparrow$.

12.7 DÉPARENTHÉSAGES

En toute rigueur, la somme de a, b, c, d devrait se noter par exemple ((a + b) + c) + d.

Les écritures rigoureuses sont généralement lourdes, et divers systèmes de déparenthésages se sont instaurés. Leur emploi n'est pas toujours clairement documenté et peut mener à des ***erreurs d'interprétation.***

12.7.1 Associativité générique

Pour un opérateur binaire $\Diamond$ associatif, la position des parenthèses n'importe pas, et on écrit souvent $x\Diamond y\Diamond z$ plutôt que $(x\Diamond y)\Diamond z$ ou $x\Diamond(y\Diamond z)$.

De proche en proche, on est ainsi conduit à noter a + b + c + d la somme de a, b, c, d, l'associativité garantissant que la signification ne dépend pas de l'ordre d'évaluation.

De même peut-on écrire pour un opérateur associatif « op », a op b op c op d au lieu de a op(b op(c op d)), voire op(a, b, c, d, ...). L'associativité conduit en effet naturellement à des *opérateurs étendus à des collections,* ou encore à des ***opérateurs à arité variable*** (cf. Lisp, occam...).

12.7.2 Pseudo-associativités

Par analogie à ce que nous venons de voir pour les opérateurs associatifs, on cherche souvent à alléger l'écriture d'expressions utilisant des opérateurs *non* associatifs. Ainsi écrivons-nous $a - b - c$ pour $(a - b) - c$, alors même que $(a - b) - c \neq a - (b - c)$.

Il s'agit alors d'une *pure convention,* qui pose un problème de consensus. En effet, convenir que $a - b - c$ signifie $(a - b) - c$, c'est dire que l'évaluation procède obligatoirement de gauche à droite.

DÉFINITION 12.22 – On parle d'***associativité à gauche*** (ou d'***évaluation de gauche à droite***) si on convient d'admettre que $x \Diamond y \Diamond z \Diamond t = ((x \Diamond y) \Diamond z) \Diamond t$.

EXEMPLES. On convient généralement d'interpréter :

- a – b – c – d comme ((a – b) – c) – d,
- a / b / c / d comme ((a / b) / c) / d.

DÉFINITION 12.23 – On parle d'***associativité à droite*** (ou d'***évaluation de droite à gauche***) si on convient d'admettre que

$$x \Diamond y \Diamond z \Diamond t = x \Diamond (y \Diamond (z \Diamond t))$$

EXEMPLES. Cette convention assez rare se rencontre

- en APL, où a – b – c – d signifie a – (b –(c – d)),
- pour certains opérateurs comme l'exponentiation ; ainsi, on interprète généralement $a \uparrow b \uparrow c$ comme $a \uparrow (b \uparrow c) = a^{b^c}$ et non comme $((a \uparrow b) \uparrow c) = a^{b \cdot c} = a \uparrow (b \cdot c)$.

NOTES

- En Prolog ISO, l'introduction d'un nouvel opérateur suppose d'indiquer le cas d'associativité (total, gauche, droite, sans), le premier cas devant être établi, les deux cas gauche/droite étant de pure convention.
- Des incertitudes sur ces questions ont rendu faux plus d'un compilateur.

12.7.3 Priorités entre opérateurs

Ayant donné un sens aux écritures allégées a + b + c + d puis a – b – c – d, on s'est posé la question d'en donner un à a + b × c – d, *qui n'en a pas a priori.*

Si l'on restitue les parenthèses absentes, 5 interprétations sont possibles.

((a + b) × c)– d	évaluation de gauche à droite, sans priorité
(a + (b × c))– d	évaluation de gauche à droite, × prioritaire sur +
(a + b) × (c – d)	+/– prioritaires sur ×
a + ((b × c) – d)	évaluation de droite à gauche, × prioritaire sur –
a + (b × (c – d))	évaluation de droite à gauche, sans priorité

L'interprétation (a + (b × c)) – d *est purement conventionnelle,* et est choisie contre ((a + b) × c) – d en *décrétant la multiplication* ***prioritaire*** *sur l'addition.*

La notion de priorité allège donc les écritures ; *mais l'échelle de priorité, sous-jacente à la syntaxe*[3]*, peut être propre à une école, une théorie, un langage, un produit.*

Pour être pratique, l'échelle de priorité doit être étendue, d'une dizaine de niveaux (Fortran, PL1, Algol 68) à 3000 (Quintus Prolog). Une échelle plus réduite peut poser des problèmes, comme en Pascal, où elle entre en conflit avec l'usage courant.

En cas de doute, on se rappellera que *le parenthésage est toujours plus sûr.*

12.7.4 Notations polonaises

Notation polonaise ou préfixée

Vers 1920, les logiciens polonais manipulaient dans leurs travaux un nombre croissant d'opérateurs logiques sans qu'il soit possible de dégager un consensus sur des conventions pour alléger les écritures. LUKASIEWICZ a montré alors qu'une notation totalement PRÉfixée, sans parenthèse ni priorité, ne comporte aucune ambiguïté si l'arité de chaque opérateur est fixe.

DÉFINITION 12.24 – Une ***expression préfixée*** est

- soit une constante,
- soit une variable,
- soit un opérateur préfixé, d'arité k, suivi de k expressions préfixées.

Ainsi, (a + b)/(c + d ↑ (k · f – u)) se transcrit : / + a b + c ↑ d – · k f u

[3] Comme le montrent les grammaires à 2 niveaux appliquées à Algol 68 ou Quintus-Prolog ou les Ampères : la notion d'*expression ou d'opérateur de niveau donné* remplace avantageusement un vocabulaire foisonnant expliquant l'expression à partir de termes, facteurs, primaires, secondaires, tertiaires, et autres concepts similaires multipliés pour concrétiser les divers niveaux de priorité, qui d'ailleurs résisteront mal aux créations d'opérateurs.

REMARQUES

- Nous retrouvons le principe de cette notation préfixée dans des phrases comme : *ranger* dans X *la somme de* b et du *quotient de* a par h, qu'on coderait ←*X* +b /a h.
- C'est la convention de base du langage LOGO.
- Sous une forme étendue à l'arité variable pour bénéficier des associativités, c'est la convention de base en LISP[4] ; en occam, c'est la convention de structuration des processus.

EXEMPLES

- (* (+ 1 2 3) (+ 4 5) (+ 6 7 8))
  ; dénote en LISP l'expression usuelle (1 + 2 + 3)·(4 + 5)·(6 + 7 + 8)
  ; les parenthèses autorisant aux opérateurs préfixés une arité variable,
- -- en occam, l'indentation définit les portées, et

```
SEQ
    PAR
        P1
        P2
        P3
    PAR
        P4
        P5
    P9
```

-- décrit un processus global anonyme exécutant en séquence

-- un premier processus anonyme regroupant 3 sous-processus concurrents
-- p1, p2, p3

-- puis un second processus anonyme regroupant 2 sous-processus concurrents
-- p4 et p5

-- puis p9.

Notation polonaise inverse[5] ou notation postfixée

De même, on peut dénoter strictement des expressions avec une ***notation POSTfixée***, sans parenthèse ni priorité, qui ne comportera aucune ambiguïté dès lors que l'arité de chaque opérateur est fixe.

DÉFINITION 12.25 – Une ***expression postfixée*** est formée

- soit d'une constante,
- soit d'une variable,
- soit de k expressions POSTfixées suivies d'un opérateur d'arité k.

Alors, (a + b) / (c + d↑(k · f – u)) se transcrit: $\boxed{a\ b\ +}\ \boxed{c\ d\ k\ f\ \cdot\ u\ -\ \uparrow\ +}\ /$

REMARQUES

- Une telle approche est sous-jacente à certaines langues comme le turc, et à certains langages comme FORTH.

---

[4] On parle alors de « notation préfixée de Cambridge ».

[5] En anglais, *Reverse Polish Notation (RPN)*.

- C'est la convention de base de certaines calculettes, des processeurs à structure de pile, et de nombreux interprètes.

Depuis 1965 environ, on utilise le procédé de transcription suivant :

Transcription Postfixée d'une Expression Parenthésée

pour chaque unité lexicale de l'expression à transcrire,
si l'unité lexicale courante est

une variable ou une constante		la transcrire en sortie, passer à l'unité lexicale suivante
une parenthèse gauche		l'empiler dans la PILE DES OPÉRATEURS, passer à l'unité lexicale suivante
un opérateur moins prioritaire que l'opérateur au sommet de la pile des opérateurs,		dépiler en sortie l'opérateur en sommet de pile
un opérateur prioritaire sur le sommet de la pile des opérateurs,		empiler cet opérateur, passer à l'unité lexicale suivante
une parenthèse droite	et s'il y a en sommet de pile	
	une parenthèse gauche	> on l'efface, on passe à l'unité lexicale suivante
	un opérateur quelconque	> on le dépile en sortie, on passe à l'unité lexicale suivante
	rien	> c'est une erreur

fin Transcription Postfixée d'une Expression Parenthésée

EXEMPLE. La transcription de l'expression parenthésée *(a+b–c)/(d–e/17)* donne successivement les états du tableau 12-1. On obtient bien finalement en sortie

$$a\ b + c - d\ e\ 17 / - /$$

REMARQUES. Pour l'expression a + b + c + d

- Si la transcription dépile les opérateurs de priorité *strictement* supérieure à celle de l'opérateur courant, on obtient l'***expression postfixée au plus tard*** :

  a + b + c + d ⇒ a b c d + + +

  a ↑ b ↑ c ↑ d ⇒ a b c d ↑ ↑ ↑

  seule écriture convenant aux opérateurs associatifs *à droite seulement*, tels que l'exponentiation.

- Si la transcription dépile les opérateurs de priorité supérieure *ou égale* à celle de l'opérateur courant, on obtient l'***expression postfixée au plus tôt*** :

  a + b + c + d ⇒ a b + c + d +
  a – b – c – d ⇒ a b – c – d –
  a / b / c / d ⇒ a b / c / d /

  seule écriture convenant aux opérateurs associatifs *à gauche seulement*, tels que la soustraction ou la division.

Pour les expressions utilisant plusieurs fois consécutives le même opérateur associatif, il y a donc 2 possibilités, marquées par la propriété :

PROPOSITION 12.12 (Landin, 1962) – La notation *postfixée au plus tôt* minimise le nombre d'emplacements nécessaires au stockage des résultats intermédiaires lors de l'évaluation d'une *chaîne associative.*

Evaluation d'une expression postfixée

Une forme postfixée s'évalue à l'aide d'une pile d'évaluation :

- un opérande en entrée entraîne l'empilement de sa valeur,
- un opérateur en entrée
  - dépile le nombre de valeurs qu'exige son arité,
  - calcule puis empile son résultat.

L'évaluation de l'expression postfixée

a b + c – d e 17 / – /

est détaillée dans le tableau 12.2.

Ce processus d'évaluation peut être adapté à la génération d'un code objet,

$Z \leftarrow (a+b-c) / (d- e/17)$ devenant

$Z\ a\ b + c - d\ e\ 17 / - / \leftarrow$

puis un flot d'instructions (fig. 12.1).

Tableau 12.1 Transcription d'une expression parenthésée en expression postfixée.

en entrée	en pile	en sortie
	[	
(	[(	
a		a
+	[(+	
b		b
–	[(–	+
c		c
)	[	–
/	[/	
(	[/(	
d		d
–	[/(–	
e		e
/	[/(–/	
17		17
)	[/(–	/
	[/	–
finExpr	[	/

Tableau 12.2 Evaluation de l'expression postfixée a b + c – d e 17 / – /.

entrée	pile niveau 0	1	2	3
a	val(a)			
b	val(a)	val(b)		
+	val(a)+val(b)			
c	val(a)+val(b)	val(c)		
–	val(a)+val(b)–val(c)			
d	val(a)+val(b)–val(c)	val(d)		
e	val(a)+val(b)–val(c)	val(d)	val(e)	
17	val(a)+val(b)–val(c)	val(d)	val (e)	17
/	val(a)+val(b)–val(c)	val(d)	val(e)/17	
–	val(a)+val(b)–val(c)	val(d)–(val(e)/17)		
/	(val(a)+val(b)–val(c)) / (val(d)–(val(e)/17))			

entrer	adr Z	↓ @ Z
entrer	val a	↓ v a
entrer	val b	↓ v b
additionner		+
entrer	val c	↓ v c
soustraire		–
entrer	val d	↓ v d
entrer	val e	↓ v e
entrer	17	↓ 17
diviser		/
soustraire		–
diviser		/
ranger		←

Fig. 12.1 Transcription d'une expression postfixée en code pour machine à pile.

Le code de la figure 12.1 suppose une *machine à (structure de) pile*, matérielle (KDF9, Electrologica X8, Burroughs 5000/5500/6500, HP3000), microprogrammée (calculette) ou le plus souvent logicielle (interprètes divers).

Ce procédé est adaptable aux machines RISC (CDC 6600 et seq., stations de travail, processeur Crusoé), en gérant l'ensemble des registres comme une pile de taille limitée.

12.8 PROPRIÉTÉS INTER-OPÉRATEURS

12.8.1 Distributivités

Elles généralisent les notions classiques de *factorisation* et de *développement*.

DÉFINITION 12.26 – Soit un ensemble E muni de deux opérations $\oplus$ et $\otimes$. L'***opérateur binaire*** $\otimes$ sera dit :

- ***distributif à gauche par rapport à*** $\oplus$ ssi $\forall x,y,z\ x\otimes(y\oplus z) = (x\otimes y) \oplus (x\otimes z)$,
- ***distributif à droite par rapport à*** $\oplus$ ssi $\forall x,y,z\ (y\oplus z)\otimes x = (y\otimes x) \oplus (z\otimes x)$,
- ***(bi-)distributif par rapport à*** $\oplus$ ssi il est distributif à droite et à gauche par rapport à $\oplus$.

EXEMPLES

- dans $\mathbb{N}$: pgcd — distributif par rapport à ppcm (et réciproquement)
- dans $\mathbb{R}$: × — distributif par rapport à + et –,
  + — distributif par rapport à max et min
  max — distributif par rapport à min (et réciproquement)
- dans $\mathbb{B}=\{0,1\}$: et — distributif par rapport à ou et ouExclusif
  ou — distributif par rapport à et.

PROPOSITION 12.13 – Si $\otimes$ est distributif par rapport à $\oplus$ et $\oplus$ associatif, on a :

$$(a1 \oplus a2) \otimes (b1 \oplus b2) = (a1 \otimes b1) \oplus (a1 \otimes b2) \oplus (a2 \otimes b1) \oplus (a2 \otimes b2)$$
$$(a \oplus b) \otimes (a \oplus b) = (a \otimes a) \oplus (a \otimes b) \oplus (b \otimes a) \oplus (b \otimes b)$$

12.8.2 Absorption

DÉFINITION 12.27 – Soit un ensemble E muni de deux opérations binaires $\downarrow$ et $\uparrow$; l'opérateur $\downarrow$ est dit

- ***absorbant à gauche pour*** $\uparrow$ ssi $\forall x,y,z\ \ x \downarrow (x\uparrow y) = x$,
- ***absorbant à droite pour*** $\uparrow$ ssi $\forall x,y,z\ \ (x\uparrow y) \downarrow x = x$,
- et ***(bi-)absorbant pour*** $\uparrow$ s'il est absorbant à droite et à gauche.

EXEMPLES

- Union/intersection, et/ou, max/min, pgcd/ppcm sont mutuellement absorbants.
- Soit l'expression booléenne ab + a'c ;
  par absorption, ab = ab + abc et a'c = a'c + a'bc,
  donc $ab + a'c$ = (ab + abc) + (a'c + a'bc)
  par commutativité et associativité = ab + a'c + (abc + a'bc)
  par distributivité = ab + a'c + (a + a')bc
  par définition du complément = $ab + a'c + bc$
  bc est dit *consensus* de ab et a'c ; c'est l'origine d'une des grandes méthodes de réduction des fonctions booléennes (voir § 15.6.3).

12.9 MORPHISMES

12.9.1 Morphisme

DÉFINITION 12.28 – Soient un ensemble E muni d'une opération binaire ◊, un ensemble F muni d'une opération binaire ♦, f une application de E dans F ; f sera dit ***morphisme*** de E dans F ssi

$$\forall x, y \in E \; f(x \lozenge y) \equiv f(x) \blacklozenge f(y)$$

On note f : <E, ◊> → <F, ♦ > un tel morphisme.

EXEMPLES

- log : $<\mathbb{R}^+, \cdot> \rightarrow <\mathbb{R}, +>$ car log(x·y) = log(x) + log(y) (Néper)
- – : $<\mathbb{R}, \min> \rightarrow <\mathbb{R}, \max>$ car – min(x, y) = max(–x, –y)
- non : $<\mathbb{B}$, et> → $<\mathbb{B}$, ou> car non(et(x, y)) = ou(non(x), non(y))
  Occam/De Morgan
- longueur : <A*, &> → $<\mathbb{N}$, +> car longueur(chaine1 & chaine2) = longueur(chaine1) + longueur(chaine2)
- mod_k : $<\mathbb{Z}, \text{op}> \rightarrow <\mathbb{Z}/k, \text{op}_k>$ car (x op y) modulo k = (x modulo k) op_k (y modulo k)

REMARQUES

- L'application f peut prendre la forme d'un opérateur unaire, et l'un et/ou l'autre opérateur, la forme d'une fonction d'arité 2.
- ***Homomorphisme*** est une ancienne désignation des morphismes.
- Un morphisme permet de procéder indifféremment à l'évaluation dans E suivi du passage à F, ou au passage à F suivi d'une évaluation dans F.
- Les représentations simplifiées utilisées pour des *preuves* ou des *contrôles* sont généralement garanties par l'existence d'un morphisme entre :
  – le calcul à contrôler, sa ou ses opérations, présumées réalistes ;
  – le calcul de contrôle, sa ou ses opérations, présumées économiques.
- La traditionnelle « preuve par P » d'un calcul arithmétique consiste à l'exécuter modulo P : le résultat du calcul modulo P (présumé simple) doit être à chaque instant égal au reste modulo P du calcul à contrôler (présumé lourd et cher).

- La vérification sémantique d'un programme à vocation physique peut simuler les calculs par une évaluation symbolique, où chaque grandeur physique est représentée par sa seule signature dimensionnelle : on contrôle ainsi, de façon générique, la validité qualitative, au sens de l'analyse dimensionnelle, d'une séquence ou d'une procédure (cf. langages Ampères).
- Dans une catégorie donnée, admettant A et B comme objets, on note **Mor**(A,B) l'ensemble des morphismes appliquant A dans B.

12.9.2 Epimorphisme

DÉFINITION 12.29 – Soit h : $<E,\Diamond> \rightarrow <F,\blacklozenge>$ un morphisme. *Si h(E) = F, le morphisme est surjectif, F est l'**image homomorphe*** de E, et h est dit ***épimorphisme.***

12.9.3 Isomorphisme

PROPOSITION 12.14 – Soit $h:<E, \Diamond> \rightarrow <F,\blacklozenge>$ un ***morphisme bijectif.*** Alors, h^{-1} est aussi morphisme bijectif.

PREUVE. Soit $h:<E,\Diamond>\rightarrow<F,\blacklozenge>$ un morphisme bijectif.

h est alors inversible, et	$\forall x, y \in E$	$h(x\Diamond y) = h(x)\blacklozenge h(y)$
donne	$\forall x, y \in E$	$h^{-1}(h(x\Diamond y)) = h^{-1}(h(x)\blacklozenge h(y))$
soit	$\forall x, y \in E$	$x\Diamond y = h^{-1}(h(x)\blacklozenge h(y))$;
qu'on transcrit	$\forall t, u \in F$	$h^{-1}(t\blacklozenge u) = h^{-1}(t) \Diamond h^{-1}(u)$

car h étant bijectif, il est équivalent de dire que chaque x de E correspond à un et un seul t = h(x) de F, ou bien qu'à chaque t de F correspond un et un seul $x = h^{-1}(t)$ de E.

Donc, h^{-1} est aussi morphisme bijectif. ■

DÉFINITION 12.30 – On appelle ***isomorphisme***, et on note $h : <E, \Diamond> \leftrightarrow <F,\blacklozenge>$ tout *morphisme bijectif* $h : <E,\Diamond> \rightarrow <F,\blacklozenge>$.

DÉFINITION 12.31 – Soit $h : <E, \Diamond> \rightarrow <F,\blacklozenge>$ un ***morphisme injectif.*** Il n'y a alors isomorphisme qu'entre E et $G = h(E) \subseteq F$. On parle alors d'*isomorphisme* de E *sur* G, ou de façon moins précise, de E *dans* F.

REMARQUE. Les ***représentations parfaites*** sont généralement garanties par l'existence d'un *isomorphisme* entre l'ensemble à représenter et l'ensemble de représentation.

C'est le cas pour les codes entiers des jeux de caractères limités tels que les jeux ISO 8859x.

C'est ainsi que les logarithmes substituent de façon réversible des calculs additifs aux calculs multiplicatifs, selon le schéma de principe : $\log: <\mathbb{R}^+,\times> \leftrightarrow <\mathbb{R}, +>$.

Mais la finitude des représentations oblige à des entorses s'il faut représenter finiment les éléments d'un ensemble non dénombrable. Ainsi, la représentation virgule flottante n'est pas isomorphe des réels : dans le domaine de définition, à un réel correspond un nombre virgule flottante le représentant, mais à ce nombre virgule flottante correspond un *intervalle* des réels représentés, englobant le réel d'origine, intervalle dont la largeur dépend de la précision de représentation. Pire, quelques cas rendent l'addition virgule flottante non associative... Une bonne représentation virgule flottante n'est ainsi qu'une moins mauvaise représentation de réels à grande dynamique.

12.9.4 Endo- et automorphisme

DÉFINITION 12.32 – Dans le cas particulier où E = F, un morphisme devient ***endomorphisme,*** un isomorphisme devient ***automorphisme.***

12.10 FORMES NORMALES & CANONIQUES

Des structures algébriques même assez légères, du fait de leurs opérateurs et de leurs propriétés, donnent naissance à une foule d'expressions dues à deux phénomènes distincts :

1. la puissance effective du système étudié ;
2. du fait des propriétés des opérateurs, les nombreuses formulations qu'on peut donner d'une même expression.

Si (1) est intéressant, (2) constitue plutôt un rideau de fumée, et si l'on ne prend aucune mesure, le formulaire à exploiter deviendra énorme, et le moindre calcul littéral, formidable.

L'automatisation des calculs littéraux[6] pousse à poser un principe d'unification des formules équivalentes par le biais d'une commune représentation.

DÉFINITION 12.33 – Soit F un ensemble de formules algébriques, et $E = \{\approx_i\}$ une collection d'équivalences élémentaires sur F. On notera $\approx$ une ***équivalence générale*** ainsi définie dans F :

1. si $f = g$ alors $f \approx g$;
2. $f \approx g$ s'il existe un i tel que $f \approx_i g$;
3. $x \approx x'$, $y \approx y'$ ... entraînent $f(x, y) \approx f(x', y'\ldots)$;
4. $f \approx g$ et $g \approx h$ entraînent $f \approx h$.

On appellera ***ensemble de formes normales sur*** F un ensemble $\Phi \subset F$ tel que :

$$\forall f \in F\ (\exists!\ \varphi \in \Phi\ f \approx \varphi\ )^7$$

Pour passer d'une formule à sa forme normale, la collection $E = \{\approx_i\}$ d'équivalences élémentaires sur F propose des règles de transformation. En général :

$$f \approx_i g \Rightarrow (f \rightarrow g, g \rightarrow f)$$

Une des transformations possibles pourra être abandonnée en fonction des propriétés recherchées pour les formes normales.

PROPOSITION 12.15 – Soit t une transformation associant à toute formule f une formule équivalente $t(f) \in \Phi \subset F$; t(f) est ***forme normale*** de f, et

- $f \approx g$ ssi $t(f) \approx t(g)$;
- en particulier, $t(f) = t(g)$ entraîne que $f \approx g$.

PROPOSITION 12.16 – *Soit t une transformation associant une même formule $t(f) = \varphi$ à toutes les formules f d'une même classe d'équivalence pour $\approx$*, alors φ est dite ***forme canonique*** des formules de la classe, et

- φ est le représentant de f pour $\approx$;
- si Φ ne contient que des formes canoniques, Φ est isomorphe à $F/\approx$;
- $f \approx g$ *ssi* $t(f) = t(g)$;
- $t(t(f)) = t(\varphi) = t(f)$ puisque $\varphi \approx f$, et, *partant de f, φ est point fixe pour t*[8].

Ces transformations t pourront :

---

[6] Y compris l'optimisation et/ou la parallélisation de programmes.

[7] $\exists!$ pour « il existe un et une seul ».

[8] Ce qu'on pourrait noter $\phi = t^{\oplus}(f)$.

- éliminer les opérateurs « secondaires » au profit des opérateurs « fondamentaux » qui suffisent à les décrire, en essayant de favoriser les opérateurs associatifs ; ex : $a - b \to a + (-b)$;
- utiliser associativité et commutativité des opérateurs fondamentaux et l'ordonnancement lexicographique des termes dans les chaînes associatives, pour garantir l'unicité de représentation entre formes similaires à des permutations près ;
  ex : $(a - b) + c + (b - a) \to (a + (-b)) + c + (b + (-a)) \to a + (-b) + c + b + (-a) \to a + (-a) + b + (-b) + c \to 0 + 0 + c$;
- évaluer les sous-expressions constantes, et exploiter les éléments remarquables ; ex : $0 + 0 + c \to 0 + c \to c$.

Ces traitements diffèrent des réductions usuelles, car ici le but est d'obtenir des formes simples et régulières facilitant les traitements ultérieurs – tels que les comparaisons d'expressions – plutôt que des expressions élégantes et compactes.

EXEMPLES

- Pour un système manipulant des polynômes, avec produits et sommes associatifs et commutatifs, une *forme canonique* pourra être un développement en *somme de monômes* dotés d'un coefficient signé non nul, les monômes étant énoncés dans l'ordre lexicographique[9] des puissances, les variables de chaque monôme étant elles-mêmes dans l'ordre alphanumérique. Ainsi,
  - $a \cdot x^2$ précède $x^2 \cdot y$ qui précède $x \cdot y^2$;
  - x^2 se note $1 \cdot x^2$, et $0 \cdot x^2$ peut être remplacé par 0 ;
  - $\varphi(x^2 + 27x^2 - 17x^2) = \varphi(1 \cdot x^2 + 27 \cdot x^2 + (-17)x^2) = \varphi((1 + 27 + -17) \cdot x^2) = 11 \cdot x^2$;
  - $\varphi(x^2 - (x + y) \cdot (x - y)) = \varphi(1 \cdot x^2 - (1 \cdot x^2 + (-1)x \cdot y + 1 \cdot y \cdot x + (-1)y \cdot y)) = \varphi(x^2 + (-1)x^2 + 1 \cdot x \cdot y + (-1)x \cdot y + 1 \cdot y^2) = \varphi(0 \cdot x^2 + 0 \cdot x \cdot y + 1 \cdot y^2) = 1 \cdot y^2$;
  - de même : $\varphi((y - 1) \cdot (y + 1) + 1) = \varphi(1 + (1 \cdot y^2 + 1 \cdot y + (-1) \cdot y + (-1) \cdot 1)) = \varphi(1 + (1 \cdot y^2 + 0 \cdot y + (-1))) = \varphi(1 + 1 \cdot y^2 + (-1)) = \varphi(1 + (-1) + 1 \cdot y^2) = \varphi(0 + 1 \cdot y^2) = 1 \cdot y^2$;

  d'où l'équivalence des deux formules : $x^2 - (x+y) \cdot (x-y) = (y-1) \cdot (y+1) + 1$.
- l'Algèbre de Boole utilise 3 styles de formes canoniques (chap. 15).

12.11 EXERCICES

12.11.0

Dans $\mathbb{B} = \{0, 1\}$, *ou* est-il distributif par rapport au *ou exclusif* ?

12.11.1

Y a-t-il, dans $\mathbb{R}^+$, distributivité de l'exponentiation par rapport à l'addition et à la multiplication ?

12.11.2

Un langage de programmation pour physiciens autorise les opérateurs numériques // et □ définis respectivement par

$$a \,//\, b \equiv 1/(1/a + 1/b) \quad \text{et} \quad a \,\square\, b \equiv (a^2+b^2)^{1/2}.$$

Etablir les propriétés de ces opérateurs en vue d'optimiser le code objet.

---

[9] Car la somme est associative et commutative.

12.11.3

Dans $\mathbb{N}$, inventorier les propriétés algébriques des pgcd et ppcm (en s'inspirant éventuellement de celles de max et min).

12.11.4

Montrer que tout opérateur associatif, commutatif et idempotent est également distributif par rapport à lui-même.

12.11.5

Donner un algorithme de transcription des expressions parenthésées en expression préfixée, et un algorithme d'évaluation des *expressions préfixées.*

12.11.6

Montrer que pour un ensemble E muni d'un opérateur $\Diamond$ associatif, commutatif et idempotent, la relation $\angle$ définie par $x \angle y \equiv x \Diamond y = x$ est un ordre.

12.11.7

Montrer que pour un ensemble E muni des opérateurs binaires $\oplus$ et $\otimes$, si $e \in E$ est élément neutre de $\oplus$, absorbant pour $\otimes$, et si $\otimes$ est distributif par rapport à $\oplus$, alors $\oplus$ est absorbant pour $\otimes$.

12.11.8

On considère dans R un opérateur $x \blacklozenge y \equiv x \cdot y \cdot (1 - y)$.

Est-il associatif ? commutatif ? idempotent ?

A-t-il un élément neutre ? un élément absorbant ? un élément involutif ? un élément idempotent ?

12.11.9 Tribu

On considère un ensemble A ; X, Y, ... désignant certains de ses sous-ensembles, on appelle tribu T attachée à X, Y, ... le sous-ensemble de 2^A formé par toutes les combinaisons possibles des X,Y, ... à l'aide des union et différence ensemblistes.

Montrer que T(X, Y, ...)

- contient $\varnothing$,
- est stable pour l'intersection,
- forme un treillis pour $\subseteq$.

Application : avec X = {a, b, c, d, e}, Y = {d, e, f}, Z = {a, f}, donner les atomes du treillis correspondant, et montrer que ce treillis est isomorphe à $\mathbb{B}^4$.

12.12 PROBLÈMES

12.12.1

Montrer qu'on peut associer à une expression une fonction de coût.

En déduire une stratégie générale de transformation des équivalences de ce chapitre en règles de réécriture, permettant d'optimiser automatiquement ces expressions.

Comment empêcher que les commutativités engendrent des transformations cycliques ? (application : optimisation de code objet ; transformation des expressions exactes brutes en expressions exactes lisibles dans les produits genre Mathematica).

12.12.2

Dans les produits genre Mathematica, il est souhaitable d'accepter des écritures variées de la part de l'utilisateur. Mais pour des traitements efficaces, il faut utiliser un jeu interne aussi réduit que possible d'opérateurs riches en propriétés. Définir une liste de transformations permettant de ramener sans erreur un jeu (externe) d'une vingtaine d'opérateurs et fonctions standard courants à un jeu (interne) d'une dizaine d'opérateurs « puissants ». (On essaiera par exemple d'éliminer de la représentation interne la soustraction, la division, la fonction abs ; de réduire le nombre de comparateurs...).

12.12.3

Etablir les propriétés de la première opération qu'un épimorphisme transmet à la seconde.

12.12.4

Soit a = 1. En déduire que $a^2 - a = a^2 - 1$, que a = a + 1, et que, par conséquent, 1 = 2 (☹).

12.12.5

Avec $\mathbb{B} = \{0,1\}$ montrer que la complémentation définit un automorphisme involutif

$$<\mathbb{B}, 0, 1, +, \cdot> \leftrightarrow <\mathbb{B}, 1, 0, \cdot, +>$$

Statut des règles de De Morgan ?

12.12.6

Dégager le système de priorités employé pour l'écriture des expressions booléennes en Fortran.

12.12.7

Dégager le système de priorités employé dans les expressions arithmétiques du langage de votre choix. L'étendre à l'affectation puis au séquenceur (opérateur de mise en séquence de deux instructions successives). Traiter le cas des parenthèses **begin...end** (resp. {}).

12.13 PISTES DE RÉFLEXION

12.13.0

Pour que la compilation d'un langage à opérateurs puisse exploiter des règles génériques d'optimisation, définir et garnir une table rectangulaire formée d'une ligne par opérateur et d'une colonne par propriété ou élément remarquable.

On décide de rendre le langage extensible, l'utilisateur pouvant créer ses propres opérateurs. Peut-on étendre sans faute l'optimisation à ces opérateurs ? quelles seraient les particularités à déclarer ?

Comment étendre le dispositif à la distributivité ? à l'absorption ?

12.13.1 Algèbres

On considère l'algèbre formel A = <S, F> formé

- d'une liste S d'ensembles ou sortes,
- d'une liste F de fonctions et d'opérateurs, munis de leurs profils, supposés relatifs aux seuls ensembles de S.

On s'intéresse à des conditions générales de viabilité d'une telle structure.

On appellera CARTE de A un graphe biparti comprenant :

- dans une classe, un sommet par ensemble ou sorte de S,
- dans l'autre, un sommet par fonction ou opérateur de F,
- un arc de chaque sorte vers chaque opérateur, chaque fois que l'opérateur utilise un argument à valeur dans cette sorte,
- un arc de chaque opérateur vers l'ensemble où son résultat prend sa valeur.

Discuter d'après cette carte des conditions de validité de l'algèbre telles que :

- aucun sommet ne doit être isolé (pourquoi?),
- la carte doit être connexe,
- la carte doit comporter au moins un circuit.

Chercher d'autres conditions.

Y a-t-il lieu de distinguer les fonctions des opérateurs ?

Peut-on distinguer simplement les cas non commutatifs ?

Faut-il introduire des opérateurs génériques standard (de construction/éclatement de tuples ? de conversion du genre reel : $\mathbb{N} \rightarrow \mathbb{R}$?)

12.13.2

Etablir les isomorphismes entre « objets » d'une même catégorie.

12.14 LECTURES

Techniques

DAVENPORT, SIRET & TOURNIER, *Calcul formel (systèmes et algorithmes de manipulation algébrique)*, Masson, coll. ERI, 1987.

DUBREIL & DUBREIL-JACOTIN, *Leçons d'algèbre moderne,* MUM n° 6, Dunod, 1964.

KAUFMANN & BOULAYE, *Théorie des Treillis en vue des applications*, chap. I-5, pp. 23-38, Masson, 1978.

MACLANE S., BIRKHOFF G., *Algèbre,* Gauthier-Villars, 1970 (2 vol.), réédition Jean Gabbay, 1996 (1 vol.).

Historiques

CALLOT, « C'était écrit ! », *Tangente,* sept/oct 1992, pp. 30-32.

DEDRON & ITARD, chap. XIII, « Les notations algébriques et les problèmes du premier degré », in *Mathématiques et Mathématiciens*, Magnard, 1959.

FREUDENTHAL H., « Notation mathématique », in *Encyclopedia Universalis,* 1999.

CHAPITRE 13

MONOÏDES & GROUPES

Le *monoïde* est la structure algébrique la plus sommaire ; elle régit par exemple l'arithmétique additive des entiers naturels. Souvent, on ne s'intéresse qu'au *groupe*, cas particulier célèbre identifié dès le début du XIXe siècle par CAUCHY et GALOIS. Cependant, la théorie des *langages formels* (et, partant, de la compilation) dérive directement de la notion de monoïde, introduite ainsi à deux titres.

13.1 PRÉSENTATION

DÉFINITION 13.1 – On appelle ***monoïde*** et on note $<A, \bullet>$ une structure formée d'*un ensemble A muni d'une opération associative* $\bullet$.

EXEMPLES

- $<\mathbb{N}, +>$.
- L'ensemble des applications de E dans E, doté du produit de composition.

13.1.1 Relation préfixe

DÉFINITION 13.2 – A tout monoïde $<A, \bullet>$ est associée une ***relation préfixe*** π définie par :

$$\forall\, x, y \in A\ x\pi y \equiv (\ \exists z \in A\ x\bullet z = y)$$

PROPOSITION 13.1 – La relation *préfixe* est transitive.

PREUVE. $x\pi y \wedge y\pi z$

$\rightarrow$ $(\ \exists t, u \in A\ (x\bullet t = y) \wedge (y\bullet u = z))$ par définition de π

$\rightarrow$ $(\ \exists t, u \in A\ (x \bullet t) \bullet u = z\ )$ en éliminant y

$\rightarrow$ $(\ \exists t, u \in A\ x \bullet (t \bullet u) = z\ )$ par associativité

$\rightarrow$ $(\ \exists w \in A\ x \bullet w = z)$ en posant $w = t\bullet u$

$\rightarrow$ $x\ \pi\ z$. ■

PROPOSITION 13.2 – La relation *préfixe* est réflexive si l'opération $\bullet$ possède un élément neutre à droite, ou si elle est idempotente.

PREUVE. π est réflexive ssi $\forall x \in A\quad (x\ \pi\ x) \Leftrightarrow \forall x \in A\ (\exists z \in A\ x\bullet z = x)$

- Si $\bullet$ possède un élément neutre à droite d,
  $\forall x \in A\ x\bullet d = x \rightarrow \forall x \in A\ (\exists z \in A\ x\bullet z = x) \rightarrow \forall x \in A\ (x\ \pi\ x)$.
- Si $\bullet$ est idempotente,
  $\forall x \in A\ x\bullet x = x \rightarrow \forall x \in A\ (\exists z \in A\ x\bullet z = x) \rightarrow \forall x \in A\ (x\ \pi\ x)$. ■

COROLLAIRE 13.3 – La relation *préfixe* est un préordre si l'opération $\bullet$ possède un élément neutre à droite, ou si elle est idempotente.

PROPOSITION 13.4 – L'éventuel *élément neutre à gauche* g de l'opération $\bullet$ *préfixe* chacun des autres éléments de A.

PREUVE. Si $\bullet$ possède un élément neutre à gauche g, $\forall x \in A\ g\bullet x = x \rightarrow \forall x \in A\ (g\ \pi\ x)$. ■

DÉFINITION 13.3 – Dans un monoïde <A, •> muni de sa relation *préfixe* :

- on appelle ***préfixe commun*** à x et y, éléments de A, un p de A tel que $p \pi x$ et $p \pi y$;
- on appelle ***préfixe commun à une partie*** $S \subseteq A$, un p de A tel que $\forall x \in S$ $p \pi x$.

PROPOSITION 13.5 – Dans un monoïde <A, •>, l'éventuel *élément neutre à gauche* g de l'opération • est *préfixe commun* (des éléments) de A (par la proposition 13.4).

EXEMPLES

- Selon Frege (≈1880), $\mathbb{N}_1 = \{1, 2, 3, \ldots\}$, peut être construit à partir de 1 et d'une addition associative, qui rend compte de ce que 5 désigne la somme de 5 uns consécutifs, le résultat ne dépendant pas des regroupements effectués :

  $1+1+1+1+1 = 1+(1+1+1+1) = (1+1)+(1+1+1) = (1+1+1)+(1+1) \ldots$

  De ce point de vue, $<\mathbb{N}_1, +>$ est un monoïde sans élément neutre, la relation π associée est l'ordre strict <, transitif et irréflexif, et en ce sens « 1 » préfixe tous les autres entiers.

- Construisons, à la Peano, $\mathbb{N}$ à partir de 0, de 1 et d'une addition associative *dont 0 est élément neutre*. Dans le monoïde $<\mathbb{N}_0, 0, 1, +>$, la relation préfixe π associée à + est maintenant réflexive : c'est un préordre, et 0 « préfixe » tous les autres. De fait, la relation π est maintenant l'ordre large ≤, de minimum 0.

13.1.2 Notation puissance

DÉFINITION 13.4 – Dans un monoïde <A, •>, on convient de noter

$$\forall a \in A \; a^m \equiv \underbrace{a \bullet a \bullet a \bullet a \ldots \bullet a \bullet a}_{\text{où a est répliqué m fois en tout.}}$$

Le résultat est en effet indépendant des regroupements :

$a^5 = a^1 \bullet a^4 = a^2 \bullet a^3 = a^3 \bullet a^2 = a^4 \bullet a^1$ peut s'écrire $a^5 = a^{1+4} = a^{2+3} = a^{3+2} = a^{4+1}$

En posant $a^1 = a$, l'associativité de l'addition des exposants reflète ainsi l'associativité du produit.

DÉFINITION 13.5 – Dans un monoïde <A, •>, on dit l'élément a ***élément k-périodique*** si $a^{k+1} = a$.

COROLLAIRE 13.6 – Tout idempotent est 1-périodique.

PROPOSITION 13.7 – Si a est k-périodique et si p, q sont deux entiers positifs congrus modulo k, alors $a^p = a^q$.

DÉFINITION 13.6 – Dans un monoïde <A, •>, on dit l'élément a ***élément finalement périodique*** s'il existe deux entiers positifs p et q > p tels que $a^p = a^q$.

REMARQUE. Pour r > q, $a^r = a^{r-q+p}$, et de proche en proche, tout a^r peut être ramené à un a^s avec $p \leq s \leq q$ et $r \equiv s \bmod (q - p)$, d'où le nom.

PROPOSITION 13.8 – Dans tout monoïde fini <A, •>, chaque élément est finalement périodique.

PREUVE. Supposons |A| = n, et soit $a \in A$. Considérons les puissances 1 à n de a :

- ou 2 de ces puissances sont égales, et l'on a $a^p = a^q$, avec $1 \leq p < q \leq n$;

- ou toutes ces puissances sont distinctes 2 à 2, et chaque valeur de A est égale à une de ces puissances ; alors a^{n+1} ne peut prendre qu'une valeur déjà rencontrée, et il existe un p tel que $1 \leq p < q \leq n + 1$. ■

EXEMPLES

- Considérons une *arithmétique archaïque* à 4 valeurs A = {1, 2, 3, beaucoup}. Elle admet une addition et une multiplication associatives. Pour cette multiplication, *1* et *beaucoup* sont 1-périodique, et *2* et *3* sont finalement périodiques.
- Les relations sur un ensemble *fini* X forment un ensemble *fini* $\mathcal{R}(X)$, que la composition dote d'une structure de monoïde ; toute relation R de cet ensemble est *finalement périodique*, au sens des puissances de composition.

PROPOSITION 13.9 – Dans tout monoïde <A, •>, chaque élément *finalement périodique* et *simplifiable* est périodique.

PREUVE. Soit a ∈ A, finalement périodique et simplifiable. 2 de ses puissances sont égales, et l'on a $a^p = a^q$, avec p < q ;

- si p = 1, a est (q–1)-périodique ;
- sinon, p > 1, et en simplifiant $a^p = a^q$ par a^{p-1}, on obtient $a = a^{q-p+1}$, et a est (q–p)-périodique. ■

13.1.3 Rôle d'un élément neutre

Notons <A, ε, •> tout monoïde <A, •> d'élément neutre ε (§ 12.4.1).

PROPOSITION 13.10 – Pour tout élément a d'un monoïde <A, ε, •>, on pose $a^0 \equiv \varepsilon$.

ARGUMENT. Pour m ∈ ℕ, on a : 0 + m = m = m + 0 . Il serait donc souhaitable que l'on ait en général :

$$\forall a \in A\ \forall m \in \mathbb{N} : a^{0+m} = a^m = a^{m+0} \text{ soit } \forall a \in A\ \forall m \in \mathbb{N} : a^0 \bullet a^m = a^m = a^m \bullet a^0$$

On a déjà $\forall a \in A \forall m \in \mathbb{N} : \varepsilon \bullet a^m = a^m = a^m \bullet \varepsilon$; il suffit donc de poser $\forall a \in A : a^0 \equiv \varepsilon$ (que rien n'interdit).

EXEMPLE. L'ensemble des applications de E dans E, doté du produit de composition, est un monoïde. I_E, l'application identique, y joue le rôle d'élément neutre, assimilable à la puissance zéro de chaque application de E dans E.

PROPOSITION 13.11 – Dans un monoïde <A, ε, •> l'*élément neutre* ε *préfixe tous les autres éléments*, en tant qu'élément neutre à gauche, *et la relation préfixe est un préordre,* du fait d'un élément neutre à droite.

13.1.4 Symétriques

Cette section s'adresse aux monoïdes <A, ε, •> dotés d'un élément neutre ε.

DÉFINITION 13.7 – On appelle ***symétrique d'un élément x pour une opération*** • un élément *x′ tel que x′• x=ε.*

EXEMPLE. Un élément neutre est toujours son propre symétrique, car ε•ε = ε.

PROPOSITION 13.12 – *La symétrisation est involutive.*

PREUVE. Si l'on s'intéresse à x′ symétrique de x, et au symétrique x″ de x′, on a : x″•x′ = ε = x′•x. Donc x″ = x″•ε = x″•(x′•x) = (x″•x′)•x [par associativité] = ε • x = x. ■

PROPOSITION 13.13 – Si x′ est symétrique de x, x est symétrique de x′ : ε = x • x′.

PREUVE. On a montré que $x'' = x$, or $x''\bullet x' = \varepsilon$ donc $x\bullet x' = \varepsilon$, ce qui symétrise la définition 13.7. ■

PROPOSITIONS

13.14 – *Tout symétrique est unique.*

PREUVE. Soient x_1 et x_2 *deux* symétriques de x . Alors :

x_1	=	$x_1\bullet\varepsilon$	ε élément neutre
	=	$x_1\bullet (x\bullet x_2)$	par définition du symétrique x_2 (prop. 13.11)
	=	$(x_1\bullet x)\bullet x_2$	par associativité
	=	$\varepsilon\bullet x_2$	par définition du symétrique x_1
	=	x_2	ε élément neutre. ■

13.15 – Tout élément a *au plus un symétrique.*

13.16 – Si un élément a possède un symétrique a′, alors l'équation $a\bullet x = b$ a pour solution $x = a'\bullet b$

PREUVE. $a \bullet x = b$ entraîne $a' \bullet b = a' \bullet (a \bullet x)=(a' \bullet a) \bullet x = \varepsilon \bullet x = x$. ■

13.2 SEMI-GROUPE

DÉFINITION 13.8 – On appelle ***semi-groupe*** un *monoïde* $<A,\bullet>$ *dont l'opération* $\bullet$ *est simplifiable*

- à droite : $\forall a,b,c \in A \quad a\bullet c = b\bullet c \rightarrow a = b$
- et à gauche : $\forall a,b,c \in A \quad a\bullet b = a\bullet c \rightarrow b = c$

i.e. dont tous les éléments sont simplifiables.

EXEMPLES

- $<\mathbb{N}, +>$ est un *semi-groupe commutatif* (Peano/Frege).
- $<\mathbb{N}, \times>$ est *un monoïde mais pas un semi-groupe,* car 0 n'est simplifiable ni à gauche ni à droite pour la multiplication ; par contre, $<\mathbb{N}_1, \times>$ est *semi-groupe.*
- $<\mathbb{N}_1$, pgcd> et $<\mathbb{N}_1$, ppcm> sont *monoïdes sans être semi-groupes* : pgcd(60,36) = pgcd(60,144) n'entraîne pas 36 = 144…

PROPOSITION 13.17 – Tout semi-groupe *fini* possède un élément neutre.

PREUVE. Le semi-groupe $<A, \bullet>$ étant fini, à n éléments, on suppose son opération définie par une table carrée n·n. L'opérateur $\bullet$ étant *simplifiable à gauche*, $\forall a, y, z \in A$ $a\bullet y = a\bullet z \Rightarrow y = z$, et par contraposition : $\forall a, y, z \in A$ $y \neq z \Rightarrow a\bullet y \neq a\bullet z$: tous les éléments de la ligne *a* sont distincts deux à deux. De même, l'*opérateur* $\bullet$ étant *simplifiable à droite*, $\forall a, y, z \in A$ $y\bullet a = z\bullet a \Rightarrow y = z$, et par contraposition : $\forall a, y, z \in A$ $y\neq z \Rightarrow y\bullet a \neq z\bullet a$: tous les éléments de la colonne *a* sont distincts deux à deux. La table constitue un ***carré latin.***

Considérons la ligne a : tous ses éléments sont distincts, l'un d'eux vaut donc a, et il existe un d tel que : $a\bullet d = a$. $\forall\, b \in A$ $a\bullet b = (a\bullet d)b = a\bullet(d \bullet b)$ par associativité, et $\forall\, b \in A$ $b = d\bullet b$ du fait de la simplifiabilité à gauche ; donc d est *élément neutre à gauche* de $\bullet$. De même considérons la colonne a : tous ses éléments sont distincts, l'un d'eux vaut donc a, et il existe g tel que $g\bullet a = a$; alors $\forall\, b \in A$ $b\bullet a = b\bullet(g\bullet a) = (b\bullet g)\bullet a$ par associativité, et $b = b\bullet g$ du fait de la simplifiabilité à droite ; donc g est *élément neutre à droite* de $\bullet$. Par suite, $d = d \bullet g = g$, et le semi-groupe $<A, \bullet>$ admet un *élément neutre.* ■

13.3 GROUPE

13.3.1 Introduction

DÉFINITION 13.9 – Un ***groupe***[1] est un monoïde $G = \langle A, \varepsilon, \bullet \rangle$ dont chaque élément x possède un symétrique.

Tout sous-ensemble B non vide d'un groupe $\langle A, \varepsilon, \bullet \rangle$ est dit ***sous-groupe*** si $\langle B, \varepsilon, \bullet \rangle$ est lui-même un groupe.

REMARQUE. Noter x^{-1} le symétrique de chaque x est compatible avec les conventions antérieures. Nous aurons ainsi $x^{-1} \bullet x = \varepsilon = x^0$; par symétrie, la notation puissance s'étend aux puissances négatives.

PROPOSITION 13.18 – $(a \bullet b)^{-1} = b^{-1} \bullet a^{-1}$

PREUVE. $a^{-1} \bullet a = \varepsilon$ donc $(a^{-1} \bullet a) \bullet b = b$ et $b^{-1} \bullet ((a^{-1} \bullet a) \bullet b) = b^{-1} \bullet b = \varepsilon$; or, par associativité $b^{-1} \bullet ((a^{-1} \bullet a) \bullet b) = (b^{-1} \bullet a^{-1}) \bullet (a \bullet b)$ d'où $\varepsilon = (b^{-1} \bullet a^{-1}) \bullet (a \bullet b)$. ■

PROPOSITION 13.19 – *Tout groupe est un semi-groupe.*

PREUVE. $x \bullet a = y \bullet a$ entraîne $(x \bullet a) \bullet a^{-1} = (y \bullet a) \bullet a^{-1}$ et, par associativité $x \bullet (a \bullet a^{-1}) = y \bullet (a \bullet a^{-1})$; or $a \bullet a^{-1} = \varepsilon$ donc $x = y$, et la simplifiabilité à droite est établie. De même, $a \bullet x = a \bullet y$ entraîne $x = y$, en prémultipliant les deux membres par a^{-1}, d'où la simplifiabilité à gauche. Le groupe $G = \langle A, \varepsilon, \bullet \rangle$ étant un *monoïde simplifiable*, est donc un *semi-groupe*. ■

EXEMPLES

- $\langle \mathbb{Z}, 0, + \rangle$ est un groupe, les *entiers signés pairs* forment un sous-groupe, mais $\langle \mathbb{N}_0, 0, + \rangle$ n'est qu'un semi-groupe, et les entiers positifs impairs ne forment même pas un monoïde pour +, puisque leur ensemble n'est pas stable pour +.
- $\langle \mathbb{Z}/k, 0, + \rangle$ est le groupe additif des entiers *modulo k*.
- $\langle \mathbb{Q}-\{0\}, 1, \cdot \rangle$ *est un groupe, mais* $\langle \mathbb{N}_1, 1, \cdot \rangle$ n'est qu'un semi-groupe.
- $\langle \mathbb{R}, 0, + \rangle$ est un groupe, $\langle \mathbb{R}^n, 0^n, + \rangle$ est un groupe, l'ensemble des translations dans $\mathbb{R}^n$ est un groupe.
- Dans le plan et dans l'espace, l'ensemble des *rotations* autour d'un point 0 forme un groupe, dont l'ensemble des *symétries* par rapport à 0 est un sous-groupe.
- Dans le plan, l'ensemble des rotations autour d'un point du plan, d'un angle multiple de 15°, forme un groupe à 24 éléments, isomorphe à $\langle \mathbb{Z}/24, 0, + \rangle$.
- $\langle \mathbb{B}, \oplus \rangle$ est un monoïde particulier, d'élément neutre 0, dont chaque élément, involutif, est son symétrique : $\forall x \; x \oplus x = 0$; donc $\langle \mathbb{B}, 0, \oplus \rangle$ est un groupe isomorphe à $\langle \mathbb{Z}/2, 0, + \rangle$.

13.3.2 Equations et opération induite

PROPOSITION 13.20 – Dans tout groupe $G = \langle A, \varepsilon, \bullet \rangle$,

- l'équation $a \bullet x = b$ $(a,b \in A)$ a pour solution $x = a^{-1} \bullet b$ $(x \in A)$,
- l'équation $x \bullet a = b$ $(a,b \in A)$ a pour solution $x = b \bullet a^{-1}$ $(x \in A)$.

[1] Concept et nom dus à Evariste GALOIS, 1811-1832.

PREUVE. Tout élément a possède un symétrique : $a \bullet x = b$ entraîne $a^{-1} \bullet b = a^{-1} \bullet (a \bullet x) = (a^{-1} \bullet a) \bullet x = \varepsilon \bullet x = x$. La seconde propriété se démontre de même en postmultipliant par a^{-1}. ■

NOTE. La résolution d'équations dans un groupe suggère une seconde opération : en posant $x / y \equiv y^{-1} \bullet x$, la solution de $a \bullet x = b$ est $x = b/a$.

- *tout groupe multiplicatif suggère une division,*
- *tout groupe additif, une soustraction...*

13.3.3 Groupes monogènes et ordres

PROPOSITIONS

13.21 – Tout élément g d'un groupe $G = \langle A, \varepsilon, \bullet \rangle$ est inclus dans un sous-groupe de G.

13.22 – L'intersection de deux sous-groupes de G contenant g est un sous-groupe contenant g.

DÉFINITIONS

13.10 – Dans un groupe, le plus petit sous-groupe contenant un élément g est $U_g = \{g^n \mid n \in \mathbb{Z}\}$. Il est dit ***groupe monogène engendré par g.***

13.11 – On appelle ***ordre d'un groupe fini*** G, et on note ord G, le nombre d'éléments de ce groupe.

13.12 – Tout groupe monogène à n éléments engendré par g est dit ***groupe cyclique d'ordre n***, et g est dit ***élément d'ordre n.***

PROPOSITIONS

13.23 – Tout groupe cyclique d'ordre n est isomorphe à $\langle \mathbb{Z}/n, 0, + \rangle$.

13.24 – Tout groupe cyclique d'ordre n est isomorphe au groupe des ***racines n-ièmes de l'unité.***

EXEMPLE. Considérons dans $\mathbb{C}$ le groupe monogène multiplicatif engendré par i. Il comprend $\{i, i^2 = -1, i^3 = -i, i^4 = 1\}$: c'est le groupe des racines quatrièmes de l'unité, isomorphe à $\langle \mathbb{Z}/4, 0, + \rangle$.

13.25 – Dans un groupe cyclique d'ordre n et de générateur g, g^m a pour ordre n/pgcd(m, n).

Ainsi, dans un groupe cyclique d'ordre 18, g est d'ordre 18, g^2 d'ordre 9, g^3 d'ordre 6, g^4 d'ordre 9, g^{21} d'ordre 6...

13.26 (Euler-Lagrange) – L'ordre de tout sous-groupe d'un groupe fini divise l'ordre du groupe.

13.3.4 Composition de groupes

PROPOSITION 13.27 – Soient $\langle G_1, \varepsilon_1, \cdot \rangle$ et $\langle G_2, \varepsilon_2, \bullet \rangle$ deux groupes, et posons

$$\forall\ (x,y), (z,t) \in G_1 \otimes G_2 : (x,y) \blacklozenge (z,t) \equiv (x \cdot z, y \bullet t)$$

$\langle G1 \otimes G2, \blacklozenge \rangle$ est un groupe d'élément neutre $\varepsilon = (\varepsilon_1, \varepsilon_2)$, et ord G = ord $G_1 \cdot$ ord G_2.

EXEMPLE. Considérons les groupes à 4 éléments. Certains sont isomorphes à $\langle \mathbb{Z}/4, 0, + \rangle$, d'autres au groupe de Klein, qu'on peut voir comme le produit cartésien de deux groupes d'ordre 2.

Tableau 13.1 Principaux groupes d'ordre 4.

groupe cyclique G_4

*	ε	a	b	c
ε	ε	a	b	c
a	a	b	c	ε
b	b	c	ε	a
c	c	ε	a	b

groupe de Klein V_4

*	ε	a	b	c
ε	ε	a	b	c
a	a	ε	c	b
b	b	c	ε	a
c	c	b	a	ε

13.3.5 Groupes abéliens, commutateurs

DÉFINITIONS

13.13 – Quand son opération est commutative, un groupe est dit ***groupe commutatif*** ou ***groupe abélien***[2] .

EXEMPLES

- $<\mathbb{Z}, +, 0>$ est un *groupe abélien.*
- Les matrices régulières n×n forment un *groupe non commutatif,* dit groupe linéaire général, noté GL(n).

13.14 – Dans un groupe non commutatif $<X, \bullet>$, on appelle ***commutateur*** x d'une paire a, b tout x tel que $(a\bullet b)\bullet x = b\bullet a$.

COROLLAIRE 13.28 – Dans un groupe non commutatif $<X, \bullet>$, le commutateur x d'une paire a, b existe toujours : c'est $x = b^{-1}\bullet a^{-1}\bullet b\bullet a$.

13.4 MONOÏDE FINIMENT ENGENDRÉ

Cette section prolonge la section 3.8.

13.4.1 Concaténation

Soit A un ensemble fini.

DÉFINITIONS

13.15 – Soient A^n l'***ensemble des séquences de longueur n bâties sur*** A, et A* l'***ensemble des séquences bâties sur*** A.

- On assimile à A l'ensemble des séquences de longueur 1 bâties sur A, si bien que $x \in A \equiv x \in A^1$.
- De là, A*, l'ensemble de tous les A^n possibles, est engendré par un opérateur de ***concaténation*** ainsi défini. Si $x \in A^m \subseteq A^*$ désigne un *m*-uple bâti sur A, si $y \in A^n \subseteq A^*$ désigne un *n*-uple bâti sur A, alors $x\bullet y$, ***concaténé*** de x et y, désigne un *(m+n)*-uple tel que :
  - pour $1 \le i \le m$: $(x\bullet y)_i = x_i$
  - pour $m < i \le m + n$: $(x\bullet y)_i = y_{i-m}$.
  - $x\bullet y \in A^{m+n}$.

13.16 – A est dit ***ensemble des générateurs*** de A*.

---

[2] En l'honneur de Niels Henrik ABEL, 1802-1829.

EXEMPLE. A* est ensemble de textes (resp. de phrases, de mots, de nombres) si A est ensemble de phrases (resp. de mots, de lettres, de chiffres).

13.4.2 Propriétés

PROPOSITIONS

13.29 – Si A est fini, A* est dénombrable.

PREUVE. En mettant A en bijection avec un système de |A| chiffres autres que 0, A* se trouve mis en bijection avec un sous-ensemble des entiers écrits en base |A|+1. ■

13.30 – *La concaténation est associative.*

PREUVE. En raisonnant composante à composante, on a toujours :
$\forall x, y, z \in A^*$ $(x \bullet y) \bullet z = x \bullet (y \bullet z)$. • est associative, et $<A^*, \bullet>$ est un monoïde. ■

13.31 – Le monoïde $<A^*, \bullet>$ des séquences bâties sur un alphabet A est un semi-groupe.

PREUVE. En raisonnant composante par composante, on a toujours :
$\forall x,y,z \in A^*$ $x \bullet y = z \bullet y \rightarrow x = z$ et $\forall x,y,z \in A^*$ $x \bullet y = x \bullet z \rightarrow y = z$. La concaténation est simplifiable à gauche et à droite, donc A* est semi-groupe. ■

13.32 – Soit |x| la longueur de x, au sens où $x \in A^m \rightarrow |x| = m$. Par construction :

$$\forall x,y \in A^* \mid x \bullet y \mid = |x| + |y|$$

13.33 – Si $<A^*, \bullet>$ possède un *élément neutre ε,* c'est *la séquence vide.*

ARGUMENT. Au sens ci-dessus, l'élément neutre doit être de longueur nulle.

13.34 – $\forall x \in A^*$ $|x^m| = m \cdot |x|$ (en particulier, $x^0 = \varepsilon \rightarrow |\varepsilon| = 0$)

DÉFINITION 13.17 – Si A est un ensemble fini,

- $A^0 = \{\varepsilon\}$ sera l'ensemble formé de la séquence vide,
- A*, l'***ensemble des séquences, vide ou non, bâties sur A*** :

$$A^* \equiv \bigcup_{i=0}^{\infty} A^i = \{\varepsilon\} \cup \bigcup_{i=1}^{\infty} A^i = A^\circ \cup A \bullet A^* = A^\circ \cup A^\oplus$$

- $A^\oplus$, l'***ensemble des séquences non vides bâties sur A*** :

$$A^\oplus \equiv \bigcup_{i=1}^{\infty} A^i = A \bullet A^* = A^* - A^\circ$$

PROPOSITIONS

13.35 – $A^\oplus$ est *le plus petit ensemble stable pour la concaténation contenant A.* A* est le plus petit ensemble stable pour la concaténation contenant A, *et muni d'un élément neutre.*

13.36 – Tout x de $A^\oplus$ de longueur finie |x| est constructible à partir de A en |x|–1 opérations.

ARGUMENT. On prend x_1 dans A, on lui concatène successivement $x_2, x_3, \ldots x_n$ où n = |x|, en n–1 opérations.

13.4.3 Ordre partiel de préfixation

PROPOSITION 13.37 – La relation ***préfixe*** π attachée au monoïde finiment engendré $<A^*, \cdot>$ est un ordre large partiel.

PREUVE. La relation π est toujours *transitive* et du fait de l'élément neutre, elle est *réflexive*. Pour l'*antisymétrie* :
Par définition : $\forall x, y \in A^* \ (x\pi y \wedge y\pi x) \rightarrow (\exists z,t \in A^* \ x\cdot z = y \wedge y\cdot t = x)$.
Supposer $x \ \pi \ y$ et $y \ \pi \ x$ nous donne : $x\cdot z = y$ et $y\cdot t = x$
Eliminons y : $x = (x\cdot z)\cdot t$
Par associativité : $x = x\cdot(z\cdot t)$
et du fait de ε: $x = x\cdot\varepsilon$
Soit par *simplification à gauche* : $z\cdot t = \varepsilon$
et, pour les longueurs : $|z| + |t| = |\varepsilon| = 0$
z et t ne peuvent être que de longueur positive ou nulle : $|z| = |t| = 0$, donc $z = t = \varepsilon$ et finalement $x = y$. ■

REMARQUE. *La relation π est maintenant un ordre, mais partiel :* si ab et abc sont comparables pour la préfixation (avec ab π abc), abc et abd ne le sont pas. Mais ε, a, ab sont ***préfixes communs*** de abc et abd ; et ab est ***le plus long préfixe commun*** de abc et abd. Ce plus long préfixe commun existe toujours, même s'il se réduit parfois au préfixe trivial ε.

PROPOSITION 13.38 – Avec ***le plus long préfixe commun*** comme borne inférieure d'une paire de séquences, la préfixation dote A* d'une structure d'*inf-demi-treillis* (§ 4.5.2).

13.4.4 Ordre lexicographique

*Un ordre total sur A induit un ordre total sur A**, faisant passer par exemple d'un *ordre alphabétique* (dans A) à un *ordre associé sur les mots* (dans A*), *dit ordre lexicographique* (cf. § 4.5.5).

DÉFINITION 13.18 – Supposons A muni d'un ordre large et total $\leq_A$ et soit $<_A$ l'ordre strict associé. Dans A*, soit une relation *précède* notée $\prec$ et définie par :

$$\forall x \in A^* \quad \varepsilon \prec x$$
$$\forall x,y \in A^{\oplus} \quad x \prec y \leftrightarrow (\exists x', y' \in A \ (\exists s, q \in A^* \ x = x'\cdot s \wedge y = y'\cdot q \wedge ((x' <_A y') \vee (x' = y' \wedge s \prec q))))$$

EXEMPLE. On prend comme A l'alphabet, et comme A* l'ensemble des séquences de lettres. Cette définition nous donne

$a \prec z$	car	$a = a\cdot\varepsilon$, $z = z\cdot\varepsilon$ et $a <_A z$
$a \prec abaque$	car	$a = a\cdot\varepsilon$, $abaque = a\cdot baque$ et $\varepsilon \prec baque$
$abc \prec z$	car	$abc = a\cdot bc$, $z = z\cdot\varepsilon$ et $a <_A z$
$abc \prec abde$	car	$abc \prec abde \leftrightarrow bc \prec bde \leftrightarrow c \prec de \leftrightarrow c <_A d$

PROPOSITIONS

13.39 – La relation $\prec$ est pour A* un *ordre large, total, dit ordre lexicographique.*

13.40 – La *préfixation* est plus fine que cette *précédence* :

$$(\forall x,y \in A^* \ x \ \pi \ y \rightarrow x \prec y) \Rightarrow \pi \subseteq \prec \text{ dans } \mathcal{R}(A^*)$$

13.5 LANGAGES FORMELS

DÉFINITIONS

13.19 – Soit A un ensemble fini de symboles, appelé dans le contexte ***alphabet*** (resp. ***vocabulaire***), et soit $A^* = \langle A, \cdot \rangle$ le monoïde associé, ***ensemble des mots*** (resp. ***ensemble des phrases***) bâti(e)s sur A.

On appelle ***langage formel*** tout sous-ensemble $L \subseteq A^*$.

On note $£_A$ l'*ensemble des parties* de A^*, vu comme ***ensemble des langages bâtis sur A.***

13.20 – Soient $L_1 \subseteq A^*$ et $L_2 \subseteq A^*$. On appelle ***concaténé*** des deux langages le langage : $L_1 \bullet L_2 = \{x \cdot y \mid x \in L1 \wedge y \in L2\}$.

PROPOSITION 13.41 – Soient $L_1 \subseteq A^*$, $L_2 \subseteq A^*$. et leur concaténé $L_1 \bullet L_2$.

1. $L_1 \bullet L_2 \subseteq A^*$.
2. $L^* \subseteq A^*$ pour tout monoïde $L^* = \langle L, \bullet \rangle$ bâti sur $L \subseteq A^*$.
3. $L_1 \bullet (L_2 \cup L_3) = L_1 \bullet L_2 \cup L_1 \bullet L_3$ et $(L_2 \cup L_3) \bullet L_1 = L_2 \bullet L_1 \cup L_3 \bullet L_3$
   /distributivité de la concaténation par rapport à l'union.

13.6 GRAMMAIRES

*Abordons les **langages en tant que productions** d'un système générateur*, défini a priori (modèle explicatif) ou a posteriori (modélisation).

DÉFINITION 13.21 – Soit T un ensemble fini de symboles, dit ***vocabulaire terminal.*** Soit N un ensemble fini de symboles, *disjoint de T*, et dit ***vocabulaire des notions.*** Soit V = N + T le ***vocabulaire*** tout court.

On appelle ***grammaire formelle*** un système

$$G \equiv \langle N, T, R \rangle$$

où $R \equiv \{x \rightarrow y \mid x \in N^{\oplus}, y \in V^*\}$ est dit ***ensemble des règles*** de la grammaire G :

- chaque couple $x \rightarrow y$ est dit ***règle*** de la grammaire ;
- à gauche, $x \in N^{\oplus}$ est une (concaténation de) notion(s), dite ***abstraction*** *de y*.
- à droite, $y \in V^*$ est dit développement ou ***production*** *de x*.

13.6.1 Dérivation

Soit $G = \langle N, T, R \rangle$ une grammaire formelle, et soient $x, y \in V^*$.

DÉFINITION 13.22 – *On dira y **dérivé**[3] **de x au sens de G**,* ou ***x abstraction de y au sens de G*** et on notera : $x \Rightarrow_G y$ *ssi* on est dans l'un des cas :

1. $x \rightarrow y \in R$ /amorçage
2. il existe $z, t \in V^*$ et $r \rightarrow s \in R$ tels que $x = z \cdot r \cdot t$, $y = z \cdot s \cdot t$ /développement local
3. il existe $z \in V^*$ tel que $x \Rightarrow_G z$ et $z \Rightarrow_G y$ /fermeture transitive

[3] Ou *développement de* x, ou *production de* x.

13.6.2 Langage engendré par une grammaire

DÉFINITION 13.23 – Soit G = <N, T, R> une grammaire, et un élément $\alpha \in N$.

*Le **langage** $L \subseteq T^*$ sera dit **de grammaire G et de germe**[4] **α** ssi L est la collection des séquences ou productions terminales dérivables de ce germe au sens de la grammaire retenue.* Alors :

$$L = \langle G, \alpha \rangle \equiv L = \{ x \in T^* | \alpha \Rightarrow_G x \}$$

13.6.3 Ambiguïtés, impasses, vide

DÉFINITIONS

13.24 – Une production p est dite ***production ambiguë*** si on peut la dériver d'une abstraction de deux façons différentes, ou plus.

Si une grammaire engendre une production ambiguë, alors on dit cette ***grammaire ambiguë.***

Un langage sera dit ***langage ambigu*** si toutes ses grammaires sont ambiguës.

13.25 – Une abstraction est une ***impasse*** :

- si aucune règle de dérivation ne lui est applicable,
- ou si toutes ses productions sont elles-mêmes des impasses.

Invoquer une telle notion bloque la dérivation.

13.26 – Pour la notation algébrique des productions on utilisera :

- ε pour la ***production terminale vide,*** *neutre pour la concaténation ;* elle est sans trace : toujours reconnue dans les analyses, elle est sans effet en génération ;
- ω pour les impasses, ω étant posé élément absorbant de la concaténation.

13.6.4 Exemple

On se donne : T = {a,b}, N = {α, β},

R = {0 : $\alpha \rightarrow \alpha\alpha$ 1 : $\alpha \rightarrow \beta$ 2 : $\beta\beta \rightarrow a$ 3 : $\beta\beta\beta \rightarrow b$}.

En détaillant la dérivation, on trouve d'abord : $\alpha \Rightarrow \beta^{\oplus}$; au-delà :

β	isolé, constitue une impasse (pas de règle $\beta \rightarrow \ldots$)
$\beta^2 \rightarrow a$	par la règle 2
$\beta^3 \rightarrow b$	par la règle 3
$\beta^4 \Rightarrow aa$	car $\beta^4 = \beta^2 \cdot \beta^2$
$\beta^5 \Rightarrow ab, ba$	où « , » est mis pour « ou », car $\beta^5 = \beta^2 \cdot \beta^3 = \beta^3 \cdot \beta^2$
$\beta^6 \Rightarrow aaa, bb$	car 6 = 2+2+2 = 3+3
$\beta^7 \Rightarrow aab, aba, baa$	car 7 = 2+2+3 = 2+3+2 = 3+2+2
$\beta^8 \Rightarrow aaaa, abb, bab, bba$	car 8 = 2+2+2+2 = 2+3+3 = 3+2+3 = 3+3+2 …

et L = {a, b, aa, ab, ba, aaa, bb, aab, aba, baa, aaaa, abb, bab, bba …}.

Cette grammaire est ambiguë car β^8 mène à aaaa soit par $\beta^2\beta^6$ soit par $\beta^4\beta^4$ soit par $\beta^6\beta^2$, et la production aaaa possède plusieurs abstractions ou analyses.

---

[4] Ou de *générateur, axiome* ou *racine* α.

13.6.5 Grammaires & compilation

La compilation repose sur une typologie des grammaires : pour chaque classe, les ensembles de règles sont soumis à des contraintes strictes, qui gouverneront les caractéristiques algébriques, voire linguistiques, des langages correspondants (le chapitre 16 traite le cas des expressions régulières ou langages à grammaires de Kleene).

On en déduit[5] des algorithmes d'analyse (qui bâtissent une représentation abstraite des textes reconnus conformes à la grammaire) et des algorithmes de génération (associant à une représentation abstraite un texte conforme) ; ces algorithmes sont éventuellement couplés par des algorithmes de transformation (si les langages diffèrent profondément) ou d'optimisation.

13.7 EXERCICES

13.7.0

On considère dans $\mathbb{R}$ l'opérateur $ défini par $x\$y \equiv (x+y)/2$; la structure $<\mathbb{R}, \$>$ est-elle un monoïde ?

Soit maintenant l'opérateur ♦ défini par $x \blacklozenge y \equiv (x^2+y^2)^{1/2}$; la structure $<\mathbb{R}, \blacklozenge>$ est-elle un monoïde ? ♦ est-il simplifiable ? y a-t-il un élément neutre ? un élément absorbant ?

13.7.1

On considère l'ensemble des fonctions $f: \mathbb{R} \rightarrow \mathbb{R}$. Et sur ces fonctions, on considère l'ensemble T des translations $\tau(a): f(t) \rightarrow f(t+a)$. Montrer que T, doté d'un produit de composition, est un monoïde. Le caractériser.

A quelle condition T peut-il être un groupe ?

13.7.2

Montrer qu'à toute opération associative, commutative et idempotente sur un ensemble A on peut associer un ordre et une structure d'inf-demi-treillis.

13.7.3

Montrer que pour une opération associative, tout élément ayant un symétrique est simplifiable.

13.7.4

Montrer que $\mathbb{B} = \{0, 1\}$ est monoïde pour + (ou), · (et), ⊕ (ou exclusif), mais que seul ce dernier opérateur dote $\mathbb{B}$ d'une structure de groupe.

Application à la résolution de l'équation en x : $a \oplus x = b$.

13.7.5

$\mathbb{B} = \{0, 1\}$ est-il monoïde pour les opérations NI(x,y) = NON(OU(x, y)) et ON(x, y) = NON(ET(x, y)) ?

---

[5] A l'aide par exemple du principe de descente récursive de Peter Lucas (1961).

13.7.6

Dans le plan, montrer que forment un groupe pour la composition :

- l'ensemble des homothéties de centre O et de rapport $k \neq 0$;
- l'ensemble des translations ;
- l'ensemble des rotations de centre O.

Ces groupes sont-ils commutatifs ?

13.7.7 Groupe S_n des permutations

Soit une suite de n objets distincts. On appelle permutation une transformation p notée $(i_1, i_2, \dots i_n)$ où les $i_1, i_2, \dots i_n$ sont des entiers de 1 à n tous distincts, et telle que : $p(x_1, x_2, \dots x_n) = (x_{i_1}, x_{i_2}, \dots x_{i_n})$ i.e. telle que l'objet de rang k dans la suite initiale devienne l'objet de rang i_k dans la suite permutée.

Montrer qu'il existe n! permutations possibles.

On s'intéresse à la composition de permutations, qu'on peut concrétiser en associant à chaque permutation un diagramme sagittal, et à la composition de permutations, correspondant à la mis en cascade de leurs diagrammes.

Montrer que ces permutations forment un *groupe* pour la composition.

13.7.8

Soient $<G_1, *>$ et $<G2, \bullet>$ deux groupes, et posons

$\forall (x,y), (z,t) \in G_1 \otimes G_2 : (x,y) \blacklozenge (z,t) \equiv (x*z, y \bullet t)$

Montrer que $<G1 \otimes G2, \blacklozenge>$ est un groupe. Définir son élément neutre.

Peut-on en déduire une possibilité de décomposition des groupes finis d'ordre non premier (ex : groupe d'ordre 12) ?

13.7.9

Montrer que les produits des combinaisons des puissances des permutations (12345) puis (678) forment finalement un groupe d'ordre 15.

13.7.10

On considère l'opération // : $\mathbb{R} \otimes \mathbb{R} \rightarrow \mathbb{R}$ définie par $a // b \equiv 1/(1/a + 1/b)$.

Montrer que $<\mathbb{R}, //>$ est un monoïde, muni d'un élément neutre et d'un élément absorbant.

Montrer que c'est de plus un groupe abélien.

En déduire que R//S = T a pour solution R = T//(– S).

Résoudre les systèmes d'équations :

1.1	P//Q = R/3	2.1	P // P = Q
1.2	P//P//P = R/2	2.2	Q // 1000 = S
		2.3	S // P = 500

13.7.11

Soit g un élément d'un groupe $<G, e, *>$. Montrer :

- que la suite des puissances entières positives de g forme un *sous-monoïde* de G ;
- que la suite des puissances entières, positives ou négatives de g, forme un *sous-groupe* de G ; justifier les notations telles que $g^{a \cdot p+m}$.

On appelle *ordre* k de l'élément g le plus petit entier k tel que $g^k = e$, s'il existe ; sinon, k est dit d'ordre infini. Montrer :

- que tout élément d'un ensemble G fini à n éléments est d'ordre n au plus,
- que pour tout élément d'ordre infini, le sous-groupe multiplicatif des puissances de g est isomorphe à <ℤ, +, 0>,
- que pour tout élément d'ordre k fini, le sous-groupe multiplicatif des puissances de g est isomorphe à <ℤ/k, +, 0>, groupe additif pour les entiers modulo k.

13.7.12

On se place dans un groupe <G, e, *>. Résoudre en x, en supposant le *produit non commutatif*

1. ax = b
2. axa = b
3. axab = b
4. axcx = bx
5. axcx = cbx

Que deviennent ces solutions avec un *produit commutatif ?*

13.8 PROBLÈMES

13.8.0 Groupe des isométries

On appelle isométries les transformations géométriques du plan qui ne changent ni la forme ni les mesures des objets.

Soit I l'ensemble des isométries d'un plan, qu'on munit du produit de composition noté o. Montrer

- qu'il existe une isométrie neutre,
- que toute isométrie a un inverse, et que par conséquent <I, o> est un groupe,
- que le sous-ensemble D des déplacements (i.e. des isométries qui conservent l'orientation : translations et rotations) est en fait un sous-groupe,
- que le sous-ensemble A des anti-déplacements (i.e. des isométries qui changent l'orientation : symétries et glissages) n'est pas stable pour la composition,
- que le sous-ensemble S des symétries par rapport à une droite
  - est inclus dans A,
  - constitue l'ensemble des générateurs de I,
  - que tout élément de I–S est composition de 2 à 4 éléments de S.

13.8.1 Tracés point par point

En vue d'une représentation informatique des tracés point par point, on considère, dans le style des codes de Freeman, l'ensemble D = {1, 2, 3, 4, 5, 6, 7, 8, 9} des ***codes du pavé numérique*** pour représenter 9 déplacements élémentaires dans le plan :

- 6 pour un déplacement unitaire parallèle à Ox,
- 9 pour un déplacement élémentaire le long de la première bissectrice,
- 8 pour un déplacement unitaire selon Oy,
- 7 pour la seconde bissectrice…

A partir de D, définir un monoïde des déplacements D* auquel appartiendront des séquences telles que 2221111888 .

Montrer que D* possède un élément neutre, que chacun des codes c de D possède un inverse 10–c, et que D* est ainsi un groupe, et même un groupe abélien (c'est-à-dire commutatif).

7	8	9
4	5	6
1	2	3

pavé numérique

On considère maintenant l'ensemble T = { $\underline{1}$, $\underline{2}$, $\underline{3}$, $\underline{4}$, $\underline{5}$, $\underline{6}$, $\underline{7}$, $\underline{8}$, $\underline{9}$} des *codes traçants*, représentant un pointage ($\underline{5}$) et 8 tracés élémentaires dans un plan. Définir à partir de T un monoïde T* auquel appartiendront des séquences telles que $\underline{2221111888}$, ayant valeur de tracé. Montrer que T* ne possède pas d'élément neutre, qu'aucun de ses codes ne possède un inverse, que T* n'est pas un groupe, et que la composition dans T* n'est pas commutative.

On considère enfin l'ensemble C = { 1, 2, 3, 4, 5, 6, 7, 8, 9, $\underline{1}$, $\underline{2}$, $\underline{3}$, $\underline{4}$, $\underline{5}$, $\underline{6}$, $\underline{7}$, $\underline{8}$, $\underline{9}$} des *codes élémentaires de tracé ou déplacement* dans un plan.

Définir à partir de C un monoïde C* auquel appartiendront des séquences telles que 222$\underline{111188}$833333$\underline{785}$.

Expliquer l'intérêt de séquences alternées pour la description de proche en proche d'image pixellisée, connexe ou pas, à partir d'un point de référence.

Montrer que C* possède un élément neutre, et un sous-groupe « des déplacements ».

On se propose d'optimiser les séquences de C* décrivant une image pixellisée.

Proposer une table de Pythagore pour la composition d'éléments de D montrant tous les cas où $d_1d_2 = d_3$.

On note qu'on pourrait aussi écrire $\underline{82}$ = 8$\underline{2}$, et, par là, 6$\underline{82}$ = 68$\underline{2}$ = 9$\underline{2}$. Développer cette remarque pour étendre efficacement à C tout entier la table précédente.

Peut-on dire que l'évaluation préalable d'un tracé suffit à l'optimiser ? Application à l'optimisation de $\underline{82}$6$\underline{82}$.

En utilisant éventuellement une approche géométrique, montrer que tout déplacement pur effectif équivaut à un déplacement simplifié $g^m h^n$, où g et h désignent deux directions successives dans 123698741. Comment combiner cette remarque avec les précédentes?

13.8.2 Analyse dimensionnelle (Joseph Fourier, 1820)

L'*analyse dimensionnelle* associe à toute grandeur physique une *signature dimensionnelle,* formule algébrique caractérisant la nature de la grandeur. Si L désigne une longueur, T le temps, M la masse,

- LT^{-1} caractérise une vitesse, LT^{-2} une accélération,
- MLT^{-2} une force, ML^2T^{-2} une énergie…

Montrer que ces signatures dimensionnelles forment un groupe, dont les dimensions M,L,T … sont les *générateurs.*

En compilation, on appelle *évaluation symbolique* toute pseudo-évaluation d'un programme permettant des prédictions sur son comportement à l'exécution ; cette évaluation exploite un système formel, tel que l'analyse dimensionnelle par exemple (Michel Sintzoff, Nice, 1976).

Les langages de la famille Ampère (Sidhoum *et al.*, 1990) sont des langages de programmation à orientation physique, où les variables et constantes de type *masse, longueur, puissance...* sont connues *en tant que telles.* Partant de la signature dimensionnelle des variables et constantes :

- définir à quelle condition sur sa signature est licite une opération prise dans $\{+, -, \times, \div, \max, \min, \uparrow, //, <, = \ldots\}$;
- donner à chaque fois le profil de l'opération, incluant la signature de son résultat.

On étendra le raisonnement à l'affectation et aux expressions conditionnelles. On montrera comment l'évaluation symbolique correspondante permet de *prouver à la compilation qu'un programme est physiquement admissible*

- en l'absence de procédures,
- si les procédures invoquées sont de *profil* connu.

Ampère 2 introduit des *procédures génériques* : montrer que les contraintes de cohérence entre arguments exploitent alors des identités entre (formules multiplicatives de) types.

13.9 PISTES DE RÉFLEXION

13.9.0

La connaissance de $\mathbb{N}$ mène aux monoïdes $<\mathbb{N}, +>$ et $<\mathbb{N}, \cdot>$. Pourquoi et comment s'est fait au fil des siècles le passage de ces structures aux groupes englobants $<\mathbb{Q}^+, \cdot>$ puis $<\mathbb{Z}, +>$?

13.9.1 Contrôles sémantiques de programmes écrits en un langage L

L'aborder sous l'angle d'une ***algèbre des types*** de L, dans laquelle chaque fonction, chaque opérateur possède un profil, spécifiant son applicabilité et son effet.

En choisissant comme évaluation symbolique d'un programme, la trace de son évaluation dans l'algèbre des types, montrer que les incohérences sont détectées en constatant la violation des conditions d'emploi d'un opérateur ou d'une fonction.

13.9.2 Calculs en intervalles

On s'intéresse aux calculs dans $\mathbb{IR} = \{[a\ b] | a, b \in \mathbb{R}, a \leq b\}$.

Pour tout opérateur op, on oppose une définition théorique $X \text{ op } Y = \{x \text{ op } y | x \in X, y \in Y\}$ et une définition calculable $X <\text{op}> Y = [\text{opinf}(X, Y), \text{opsup}(X,Y)]$.

L'addition entre intervalles a-t-elle un élément neutre ? est-elle associative ? est-elle commutative ? est-elle simplifiable ?

13.9.3 Calculs qualitatifs

La physique naïve pratiquée par chacun – par exemple pour envoyer une boule de papier dans une corbeille – suggère que les grandeurs n'importent que par leur ordre de grandeur. Est-ce possible ?

Supposons toute valeur v dénotée par le rang du nombre le plus proche dans une progression géométrique de référence. Supposons une raison $r > 1$: tout nombre réel positif x sera dit de *taille k* donnée par : $r^{k-1/2} \leq x < r^{k+1/2}$, et à la taille k on associera r^k, « représentant » de l'intervalle.

Montrer qu'il existe un groupe additif sur les tailles. Est-il homomorphe d'un groupe multiplicatif sur les valeurs ?

Comment additionner deux grandeurs de taille k_1 et k_2 ?

Cette addition est-elle associative ?

13.10 LECTURES

BENZAKEN C., *Systèmes formels, Introduction à la logique et à la théorie des langages,* Masson, 1991.

DELEDICQ A., *Le monde des symétries*, ACL Ed., coll. Mathématiques pour tous, vol. 2, 1993.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien*, tome I, chap. VI, Coll. Programmation, Ed. Gauthier-Villars, 1973.

GROSS M. et LENTIN A., *Notions sur les grammaires formelles,* Coll. Programmation, Ed. Gauthier-Villars, 1967.

KAUFMANN A., BOULAYE G., *Théorie des treillis en vue des applications*, chap. I, Ed. Masson, 1978.

MAC LANE S. et G. BIRKHOFF G., *Algèbre 1/Structures fondamentales,* Ed. Gauthier-Villars, 1970.

NÄGELI H.H., *Mathématiques discrètes 1*, Collection Informatique, Presses polytechniques et universitaires romandes, Lausanne, 1998 (groupes pp. 207-227).

REINHARDT F. et SOLDER H., *Atlas des mathématiques*, p. 39, p. 3 et sq, La Pochothèque, 1997.

SIDHOUM H., BABOUT M., FRÉCON L., « Ampère 2, un langage de programmation pour la physique », *The European Journal of Physics*, vol. 11, 1990, pp.163-171.

coll., *Petite encyclopédie des mathématiques*, chap. XVI & I, Edition Pagoulatos, Didier distr., 1965.

CHAPITRE 14

DIOÏDES

Cette théorie est née de la volonté d'unifier de nombreux algorithmes matriciels similaires liés à la *recherche opérationnelle,* à la *théorie des relations,* à l'*approche équationnelle de la théorie des langages*. Introduite par J. KUNTZMANN en 1972, cette théorie a été développée et diffusée par GONDRAN & MINOUX à partir de 1979.

14.1 PRÉSENTATION

DÉFINITION 14.1 – On appelle ***dioïde*** une structure $<A, \omega, \varepsilon, +, \cdot>$ formée d'un ensemble A muni de deux éléments remarquables ω et ε et de deux opérations binaires internes + et · telles que :

+ est *associative, commutative, d'élément neutre* ω ;
· est *associative, d'élément neutre* ε*, d'élément absorbant* ω,
et *distributive par rapport à* +.

EXEMPLES

- $<\mathbb{B} = \{0, 1\}, 0, 1, +, \cdot>$ est un dioïde sous-jacent à l'algèbre de Boole.
- $<\mathbb{N}, 0, 1, +, \cdot>$ est un dioïde, comme $<\mathbb{R}, 0, 1, +, \cdot>$.
- $<\mathbb{R}, -\infty, 0, \max, +,>$ est un dioïde.

CONVENTION. Une expression comme $(x\cdot y)\cdot z + t\cdot u$ se notera usuellement xyz+tu (produit « · » implicite) s'il n'y a pas d'ambiguité.

14.1.1 Propriétés

PROPOSITION 14.1 – Un *dioïde* est deux fois monoïde :

- *monoïde commutatif* pour sa première opération,
- *monoïde* pour la seconde.

PROPOSITION 14.2 – La relation *préfixe* définie dans un dioïde D par :

$$\forall x,y \in D\ (x\ \pi\ y) \equiv (\exists z \in D\ x\cdot z = y)$$

est toujours un *préordre.*

PREUVE. Elle est toujours

- réflexive du fait de l'élément neutre ε de « · »,
- transitive du fait de l'associativité de « · ». ■

14.1.2 Dioïde finiment engendré

DÉFINITION 14.2 – Soit A un *ensemble fini* et A* l'ensemble des séquences bâties par concaténation sur A ; soit A^{*+} l'ensemble des sommes de séquences sur A*.

Une notation $A^{*+} = <A, \omega, \varepsilon, +, .>$ *désignera A comme* ***ensemble des générateurs du dioïde*** A^{*+}.

Compte tenu des associativités et de la distributivité du produit sur l'addition, A^{*+} est *stable* pour les deux opérations.

14.1.3 Gerbier

DÉFINITION 14.3 – *On appellera* ***gerbier*** *un* ***dioïde idempotent****, dioïde dont la première opération est idempotente.*

PROPOSITION 14.3 – Dans un *gerbier*, il existe un ***ordre de contenance***.

PREUVE. Posons $x \ll y$, lu x *dans* y, s'il existe z tel que $y = x + z$; $\ll$ est le préordre préfixe associé à +.

Si l'on a $x \ll y$ et $y \ll x$ alors il existe z et t tels que $y = x + z$ et $x = y + t$, et :

x	$= y+t = (y + y) + t$	idempotence de +,
	$= y + (y + t)$	associativité de +,
	$= y + x$,	définition de t,
	$= x + y$	commutativité de +,
	$= x + (x + z)$	définition de z,
	$= (x + x) + z$	par associativité de +,
	$= x + z$	par idempotence de +,
	$= y$.	

L'antisymétrie du préordre $\ll$ est ainsi établie, et *$\ll$ est un ordre.* ■

PROPOSITION 14.4 – *Avec* + idempotente, *la* séquence vide *comme ε et la* séquence impossible *ou impasse comme ω, A^{*+} est isomorphe à $\pounds_A$.*

NOTE. Les A^{*+} constitueront donc un cadre pour certaines questions d'Algèbre relatives aux langages, dès lors que :

est vu comme une *concaténation* ou composition séquentielle non commutative,

\+ est une *agrégation* de séquences parallèles, coopératives ou concurrentes, dont l'idempotence, héritée de l'union ensembliste, élimine des redondances.

14.2 SEMI-ANNEAU

DÉFINITION 14.4 – Un ***semi-anneau*** est un dioïde, *simplifiable* pour sa première opération.

EXEMPLES

- $<\mathbb{N}, 0, 1, +, \cdot>$ est un semi-anneau.
- L'ensemble des *polynômes à coefficients réels positifs, et à une variable*, est un semi-anneau.

14.3 ANNEAU

DÉFINITION 14.5 – Un ***anneau*** est un dioïde, *groupe pour sa première opération*, i.e. dont chaque élément a un symétrique pour cette première opération.

EXEMPLES

- le dioïde $<\mathbb{Z}, 0, 1, +, \cdot>$ est un anneau ;
- le dioïde $<\mathbb{B} = \{0, 1\}, 0, 1, \oplus, \cdot>$ est un anneau *de Boole* ;
- l'ensemble des *polynômes à coefficients réels et à une variable,* est un anneau.

14.4 CORPS

DÉFINITION 14.6 – Un ***corps*** est un anneau <A, ω, ε, +, * >, tel que <A–{ω}, *> soit aussi un *groupe.*

*Les **corps finis** sont dits **corps de Galois***[1] .

EXEMPLES

- <ℚ, 0, 1, +, · >, <ℝ, 0, 1, +, · > sont des corps.
- L'ensemble des *quotients de polynômes à coefficients réels et à une variable*, est un corps.
- <𝔹 = {0, 1}, 0, 1, ⊕, · > est le plus petit *corps de Galois*, noté CG(2), utilisé notamment en *théorie des codes détecteurs ou correcteurs d'erreurs* (cf. W.W. PETERSON).

14.5 GRAPHE VALUÉ

14.5.1 Marquage d'un chemin et monoïde d'énonciation

DÉFINITION 14.7 – G_V = <X, U, V, μ : U→V> sera dit ***graphe marqué*** si

- X est l'ensemble des sommets,
- U est un ensemble d'arcs tels que $U \subseteq X^2$,
- μ est une ***fonction de marquage des arcs*** par un élément de V.

A tout chemin C dans G associons une séquence formée de la suite des marques rencontrées sur les arcs successifs du chemin C.

Cette séquence relève d'un monoïde V*, dès lors que la composition séquentielle est associative, i.e. que l'on peut associer à un chemin formé de trois arcs ou sous-chemins marqués m, n, p : ((m, n), p) = (m, (n, p)) = (m, n, p) noté plus simplement m·n·p voire mnp : en particulier, la concaténation des énoncés des sous-chemins est indépendante de la fragmentation utilisée.

Si de plus μ est une injection marquant chaque arc de façon unique, et si la composition séquentielle se réduit à une concaténation, toute ***énonciation*** de chemin est alors spécifique du chemin énoncé.

NOTE. On parlera plutôt de *graphe marqué* si le marquage est symbolique, et de *graphe valué* si le marquage est numérique, ou relève d'une échelle.

14.5.2 Marque d'absence

Supposons pour le graphe marqué G_V *qu'il existe une composition séquentielle associative des marquages des arcs, et un $\omega \in V$ absorbant pour cette composition.*

La fonction de marquage μ peut être maintenant définie directement comme *application* de $X^2 \to V$ si l'on convient que ω, élément absorbant, sera la ***marque d'une absence d'arc**, et plus généralement dénotera les impasses.*

On assimile ainsi arc manquant à arc (présent) interdit/impossible. D'autre part, un chemin ne menant qu'à une impasse devient lui-même impasse, que l'impasse soit initiale ou lointaine.

On dénote alors *plus simplement* G_v par : G_v = <X, V, ω, μ : $X^2 \to V$>.

---

[1] En l'honneur d'Evariste Galois, 1811-1832.

14.5.3 Graphe topologique

DÉFINITION 14.8 – *Par opposition au graphe valué, on appellera* ***graphe topologique*** le graphe simple sous-jacent $G = (X,U = \{u \in X^2 \mid \mu(u) \neq \omega\})$, qui définit *la structure sans la valuation.*

14.5.4 Faisceaux de chemins et dioïde

DÉFINITION 14.9 – Dans un graphe G donné, on appellera ***faisceau*** *de chemins* F_{xy}, une collection de chemins allant de x à y.

Pour un graphe valué $G_v = \langle X, V, \omega, \mu : X^2 \rightarrow V \rangle$, si chacun des chemins est à valeur dans un monoïde V*, la valuation d'un faisceau supposera une synthèse ou agrégation de ces valeurs, concrétisant sous la forme souhaitée les coopération et/ou compétition entre (sous-)chemins, de la simple énumération à des combinaisons plus ou moins achevées,

Pour cela, on cherchera si on peut substituer au monoïde V* de valuation des chemins, un dioïde V^{+*} dont la première opération reflèterait l'agrégation cherchée.

NOTE. Soit à rechercher un bon itinéraire dans un système routier incertain. On associe à chaque arc (liaison élémentaire) une probabilité de parcours sans incident. Cette probabilité de parcours sans incident d'un chemin sera, en première approximation, le produit des probabilités de parcours sans incident de ses composantes. Une liaison inexistante ou impossible sera assimilée à une liaison de probabilité 0.'

Si plusieurs solutions s'offrent de x à y,

- pour un véhicule isolé, on ne retient du faisceau des chemins possibles que le chemin de probabilité max ;
- pour une flotte de véhicules au-delà des capacités de ce chemin, on peut décider de retenir pour le même faisceau :
  - soit les k chemins de probabilité max,
  - soit tous les chemins, classés par probabilité décroissante...

Déterminer si le système de valuation des faisceaux constitue bien un dioïde reviendra à prouver que ce système de valuation de faisceau jouit des propriétés génériques attachées à cette catégorie. A contrario, si le système de valuation des faisceaux ne constitue pas un dioïde, les traitements, purement ad hoc, devront être définis avec prudence.

14.5.5 Matrice d'un (multi)graphe valué

La nouvelle définition autorise la représentation d'un graphe valué G par une matrice [G] à valeurs dans V, ayant dans la position (i, j) :

- la valeur ω s'il n'existe pas d'arc de x_i à x_j,
- toute valeur v de V s'il existe un arc de marque v de x_i à x_j,
- tout *somme* $v_1+v_2+v_3$... de telles valeurs, au sens de la première opération du dioïde, s'il existe de x_i à x_j *plusieurs* arcs valués respectivement v_1, v_2, v_3...

14.6 CALCULS MATRICIELS

14.6.1 Somme et composition

DÉFINITION 14.10 – Considérons 2 relations valuées $S = (X, A, \sigma : X^2 \longrightarrow A)$ et $T = (X, A, \tau : X^2 \longrightarrow A)$, et un dioïde associé $A^{+*} = \langle A, \omega, \varepsilon, +, \cdot \rangle$.

Aux matrices [S] et [T] à valeur dans A, nous associons les matrices $[S \cup T]$ et $[S \bullet T]$ à valeur dans le dioïde, sur la base des formules :

$$[S \cup T]_{ij} = s_{ij} + t_{ij}, \quad \text{où } s_{ij} = \sigma(i, j) \text{ et } t_{ij} = \tau(i, j)$$

$$[S \bullet T]_{ij} = \Sigma_k \, s_{ik} \cdot t_{kj}$$

ou, en bref,

$$[S \cup T] = [S] + [T] \quad \text{et} \quad [S \bullet T] = [S] \cdot [T]$$

Au-delà, le formalisme des dioïdes reprend les calculs des chapitres 8 et 9, sous réserve du réexamen des simplifications utilisées.

14.6.2 Matrice nulle et matrice unité

DÉFINITION 14.11 – La ***matrice nulle*** [Ω], neutre pour la somme, n'est formée que de ω.

La ***matrice unité*** [I] doit être neutre pour le produit de composition. C'est une ***matrice diagonale,*** *formée de ω, sauf la diagonale principale formée de ε. Le graphe associé se réduit à une boucle neutre par sommet.*

Tableau 14.1 Matrice unité [I].

$$\begin{bmatrix} \varepsilon & \omega & \omega & \omega & \omega \\ \omega & \varepsilon & \omega & \omega & \omega \\ \omega & \omega & \varepsilon & \omega & \omega \\ \omega & \omega & \omega & \varepsilon & \omega \\ \omega & \omega & \omega & \omega & \varepsilon \end{bmatrix}$$

14.7 THÈSE CENTRALE

Soit $G = \langle X, \mu : X^2 \rightarrow V \rangle$ un graphe valué (resp. $G = \langle X, \mu : X^2 \rightarrow V^+ \rangle$ un multigraphe valué), représentable par une matrice carrée M à valeur dans $V^{[+]}$.

Les problèmes relatifs aux chemins dans un tel (multi)graphe sont résolus par le calcul des matrices

$$M^{\oplus} = \sum_{k=1}^{\infty} M^k \quad \text{ou} \quad M_n = \sum_{k=1}^{n} M^k$$

selon qu'on s'intéresse aux chemins de taille illimitée ou limitée, calculs menés dans un dioïde V^{+*} convenable.

14.8 TAILLE ILLIMITÉE

La question cruciale ici est de justifier la substitution d'une somme finie, calculable, à la somme infinie officiellement cherchée. Cette troncature peut être légitime parce que le graphe est sans circuit (tous les chemins sont élémentaires et les puissances élevées sont alors nulles) ou du fait de la nature précise du dioïde.

14.8.1 Cas d'un gerbier fini

Adaptons ici un raisonnement de C. BERGE pour les graphes simples / relations booléennes[2].

Soit $n = |X|$ le nombre de sommets en cause, et $v = |V^{+*}|$ le nombre d'éléments du dioïde V^{+*} présumé fini.

Il existe alors au plus v^{n^2} matrices n·n distinctes à valeurs dans V^{+*}.

De ce fait, il existe p et q tels que $M^p = M^q$ avec $p < q \leq v^{n^2} + 1$.

Alors, si $r > q$, $M^r = M^q \cdot M^{r-q} = M^p \cdot M^{r-q} = M^{r+p-q}$. En répétant éventuellement le procédé, on obtient

$$M^r = M^{((r-p) \bmod (q-p))+p}$$

et la suite des M^r apparaît comme périodique, de période $q - p$, au-delà de M^p.

Du fait de l'idempotence de la somme, la première période absorbe toutes les autres. On a donc $\sum_{k=p}^{\infty} M^k = \sum_{k=p}^{q} M^k$ et, de là $\sum_{k=1}^{\infty} M^k = \sum_{k=1}^{q} M^k$ soit $M^{\oplus} = M_q$.

L'effet des chemins de taille illimitée est ici calculable en q étapes (où q est éventuellement très grand : pour $v = 5$ et $n = 10$, $q \leq 5^{100} \approx 10^{70}$).

14.8.2 Cas d'un gerbier absorbant

Supposons un dioïde idempotent, où de plus ε est absorbant pour +.

Considérons un chemin d·f formé d'un sous-chemin d suivi d'un sous-chemin f, et soit C un circuit passant par leur sommet de jonction. Alors,

$$d + d \cdot C = d\,(\varepsilon + C) = d, \text{ et } d \cdot f + d \cdot C \cdot f = d(f + C \cdot f) = d(\varepsilon + C) \cdot f = d \cdot f$$

Dans un tel cas, *la valeur attachée à tout chemin élémentaire absorbe les valeurs attachées aux chemins non-élémentaires de mêmes extrémités qui n'en diffèrent que par un ou plusieurs circuits.*

Par abus de langage, on dit que *tout chemin élémentaire absorbe les chemins non élémentaires de mêmes extrémités qui n'en diffèrent que par un ou plusieurs circuits.*

Dans ces conditions, le calcul peut se réduire aux chemins élémentaires, et $M^{\oplus} = M_{n-1}$.

14.8.3 Cas des anneaux

Par définition, nous avons toujours :

$$M^* = I + M \cdot M^* \text{ et } M^{\oplus} = M + M \cdot M^{\oplus}$$

Si le dioïde est un anneau, il est groupe pour la première opération ou somme, à laquelle on peut associer une soustraction ; alors, formellement :

$$(I - M) \cdot M^* = I \text{ et } M = (I - M) \cdot M^{\oplus}$$

Si l'anneau se trouve être un corps, et si l'inversion de la matrice (I–M) a un sens :

$$M^* = (I - M)^{-1} \text{ et } M^{\oplus} = (I - M)^{-1} \cdot M$$

---

[2] Ou disons que l'ensemble des matrices considérées étant fini, et ces matrices formant un monoïde multilplicatif, chacune est *finalement périodique* au sens de la proposition 13.6.

Lorsqu'elles sont effectives, ces remarques autorisent la reprise de procédés d'*algèbre linéaire*, dans la mesure où leurs exigences sont satisfaites [GONDRAN & MINOUX].

14.9 TAILLE LIMITÉE

Par nature ou par réduction, le problème est d'évaluer des $M_p = \sum_{k=1}^{p} M^k$.

On peut toujours les calculer comme dernier terme d'une suite finie :

$$M_1 = M$$
$$M_k = M + M \cdot M_{k-1} = M \cdot (I + M_{k-1})$$

Il y a donc (p–1) étapes comportant

- n^2 sommes au titre de la somme matricielle (réductible à n),
- $n^2(n-1)$ sommes et n^3 produits élémentaires au titre du produit matriciel.

Le coût est donc en n^3p, soit n^4 si p = n.

14.9.1 Gerbier

On a toujours

$$M_1 = M \text{ et } M_k = M + M \cdot M_{k-1}$$

qu'on peut réécrire, du fait de l'idempotence de la somme :

$$M_k = M_{k-1} + M \cdot M_{k-1} = (I+M) \cdot M_{k-1} = M' \cdot M_{k-1} \qquad \text{(i)}$$

en posant M′ = M + I.

On évite déjà une somme de matrices à chaque étape, ce qui est sensible aux faibles valeurs de n. Mais on peut aussi calculer directement (ii) $M_p = M'^{\,p-1} \cdot M$ en appliquant la méthode de Legendre pour obtenir la puissance (p – 1) de M′ nécessaire.

Le coût opérationnel est alors d'environ

- $n^3 \cdot (p - 1)$ sommes et produits élémentaires si on procède pas à pas selon (i),
- réduit à un $O(n^3 \cdot ld(p))$ si on calcule (ii) à l'aide de la méthode de Legendre (cf. sect. 6.1).

La V-réflexivité du graphe accélère le procédé.

14.9.2 Graphe V-réflexif

DÉFINITION 14.12 – Une ***relation*** S à valeur dans V, un ***graphe*** S valué dans V seront dits ***V-réflexifs*** [3] ***ssi***

$$[S] = [S] + [I]$$

Ce sera le cas de toute relation (resp. graphe) dont la matrice possède une *diagonale principale formée de ε, si ε est absorbant pour la somme du dioïde.*

Alors $S_k = S^k$, ce qui pour le calcul gagne du temps et surtout de l'espace.

---

[3] Ce point renvoie à une théorie des relations basée sur une *logique non standard* dès que |v|>2.

14.10 EXEMPLES

14.10.1 Table de distances[4]

Considérons une carte comportant les distances ville à ville pour les liaisons directes. Peut-on en déduire les distances ville à ville utilisant les liaisons les plus courtes, directes ou non ?

Analyse

- Les distances sont des nombres positifs : valeurs dans $\mathbb{R}^+$.
- La longueur d'un chemin composé est la somme des longueurs des sous-chemins. → *monoïde de base* $<\mathbb{R}^+, +>$
- Si un itinéraire peut prendre 2 chemins différents de même extrémité, je choisis le plus court → *agrégation par min*

La structure $<\mathbb{R}^+, \min, + >$ est-elle bien un dioïde ?

- min est associatif, commutatif, idempotent, d'élément neutre $\omega = +\infty$,
- « + » est associatif, d'élément neutre $\varepsilon = 0$, d'élément absorbant $\omega = +\infty$,
- enfin, pour tout a,b,c réels positifs : $\min(a,b) + c = \min(a+c, b+c)$

$$\rightarrow \mathbb{R}^{min+} \equiv <\mathbb{R}, \omega = +\infty, \varepsilon = 0, min, + > \textit{ est un dioïde,}$$

- ce dioïde est *idempotent* : c'est un *gerbier*,
- dans $\mathbb{R}^+$, $\varepsilon = 0$ *est absorbant pour min*,
- *le graphe aux distances est* $\mathbb{R}$*-réflexif* : on a toujours $d(x,x) = d[x,x] = 0 = \varepsilon$.

Tableau 14.2 Matrice initiale de distances.

de à	a	b	c	d	e	f	g	h
a	0	20	∞	∞	∞	∞	∞	50
b	20	0	25	∞	∞	∞	∞	57
c	∞	25	0	18	∞	50	48	48
d	∞	∞	18	0	75	60	63	∞
e	∞	∞	∞	75	0	70	∞	∞
f	∞	∞	50	60	70	0	80	∞
g	∞	∞	48	63	∞	80	0	30
h	50	57	48	∞	∞	∞	30	0

Donc, l'effet de tous les chemins est obtenu en se limitant aux seuls chemins élémentaires, ici à la puissance 7. Donc : $M^{\oplus} = M_7 = M^7$.

On peut, en considérant que les puissances supérieures n'apportent rien, préférer calculer $M_8 = M^8 = M^{2^3}$ par Legendre, pour un coût réduit (3 contre 6).

Pour 1000 villes ou sites, on préfèrera $M_{1024} = M^{1024} = M^{2^{10}}$ qui, suivant Legendre, réduirait le coût dans un rapport 1000/10, soit un facteur 100.

---

[4] Cas le plus favorable : somme idempotente, chemins optimaux nécessairement élémentaires.

Comment calculer la table des distances case par case ?

On utilise un *produit matriciel, paramétré en* ses données et *ses opérateurs, avec ici*

min *comme* somme *et* + *comme* produit

En vertu des considérations précédentes, la réponse s'obtiendra en élevant la matrice de départ 3 fois de suite au carré, pour atteindre les tailles 2, 4, 8.

Pour les chemins à 1 et 2 arcs, calculons la matrice carrée de la matrice d'origine, en procédant comme suit case à case.

EXEMPLE. Soit à calculer le terme $D^2(c, e)$.

On part de la *li*gne D(c, *) et la *col*onne D(*, e) de la table des distances.

Pour chaque intermédiaire x on calcule D(c, x) + D(x, e), longueur attachée au chemin (c, x, e), finie s'il existe, infinie sinon, l'élément absorbant ayant précisément été choisi pour autoriser un *calcul aveugle.*

Ensuite, l'opérateur d'agrégation *min* garde la longueur la plus intéressante.

Ici, il existe 2 chemins à 2 arcs de c à e : c-d-e et c-f-e ; le premier est le plus court.

Les solutions de longueur ∞ (ou non-solutions) sont éliminées par la première solution rencontrée.

Tableau 14.3 Longueur des chemins c-*-e de taille 2.

	a	b	c	d	e	f	g	h	
c à *	∞	25	0	18	∞	50	48	48	
* à e	∞	∞	∞	75	0	70	∞	∞	
min (c–*–e	∞	∞	∞	93	∞	120	∞	∞	) = 93

La programmation du produit de matrices, de l'élévation au carré, et finalement de l'application reste donc simple et directe.

Ne pas confondre *complexité* et *étrangeté.*

14.10.2 Autres cas de chemins optimaux

[GONDRAN & MINOUX] aborde(nt) de nombreux problèmes de cheminement optimal, dont les principaux sont donnés par la table 14.4.

Tableau 14.4 Problèmes à dioïdes.

	Eléments du dioïde				
Problèmes	A	ω	ε	+	*
chemin de capacité max	$\mathbb{R}^+$	0	+∞	max	min
chemin de taille min	$\mathbb{N}$	+∞	0	min	+
plus court chemin	$\mathbb{R}$	+∞	0	min	+
plus long chemin	$\mathbb{R}$	–∞	0	max	+
chemin le plus sûr	[0, 1]	0	1	max	

14.10.3 L'évocation en documentation automatique

Soit à accéder aux documents d'une collection D à partir de requêtes basées sur une collection M de mots-clés.

Cette activité entraîne 2 sortes de difficultés :

- il y a *silence* si le système ne retrouve pas un ou plusieurs documents qu'il aurait normalement dû associer à la requête,
- il y a *bruit* si le système indique un ou plusieurs documents qu'il n'aurait pas dû normalement associer à la requête.

Les deux ne sont antagonistes qu'en apparence : un système documentaire médiocre peut être silencieux ET bruyant si une requête mène à de nombreuses références non pertinentes en oubliant des références pertinentes.

Contre le silence on a utilisé la troncature : *cheva$* pour *cheval et chevaux,* mais aussi *chevaine, chevalet, chevalier, chevalière, chevau-léger, chevauchée...*, d'où une tendance au bruit.

On a aussi tenté de traiter la synonymie, mais faut-il admettre qu'elle soit transitive avec les quasi-synonymes ?

Ne faut-il pas aborder le problème de façon plus générale, en termes par exemple de *champs sémantiques*, pour qu'une requête telle que *élevage de chevaux* mène bien à un document parlant de *juments*, de *poulains,* d'*écuries,* de *haras,* d'*étalons, même si cheval (ou chevaux)* est absent ?

On généralise la synonymie en une relation d'*évocation* $E \subseteq M \otimes M$, associant à chaque couple $(a, b) \in E$ une *force d'évocation.* Pour cela, on décide de traiter E en relation floue, en posant de fait $E : M \otimes M \rightarrow [0, 1]$.

On aura : a E b = 1 si a évoque b *absolument*,
a E b = 0 si a *n'évoque pas* b,
a E b prenant des valeurs intermédiaires sinon.

On pourra donc avoir des *quasi-synonymes* et des dissymétries. Par exemple, on prendra

E (bœuf, bovin) = 1 pour marquer que *tout bœuf est un bovin*[5],
E (bovin, bœuf) = 0,8 pour indiquer qu'*un bovin est souvent un bœuf.*

Restent alors deux problèmes :

- trouver la force d'évocation attachée à un chemin, connaissant celle attachée à chacun de ses arcs ou sous-chemins,
- trouver la force d'évocation d'un vocable par un autre, connaissant le faisceau de chemins de l'un à l'autre ; pour cela, définir d'abord la force d'évocation attachée à une paire de chemins disjoints unissant x à y, et de forces respectives f_1 et f_2.

Typiquement, on peut envisager l'un des 3 systèmes ci-après.

---

[5] Dans un tel cas, bovin est dit *hyperonyme* et bœuf, *hyponyme* ; sans représenter tout un champ sémantique, ces dispositions reflètent tout au moins les classifications, et les proximités sémantiques correspondantes.

Tableau 14.5 Combinaison d'évocations.

Style	ω	ε	+ /agrégation	· /séquence	
Zadeh	0	1	max	min	favorise les longs chemins d'évocation
pseudo-probabiliste	0	1	⊕ où $a \oplus b \equiv a+b-a \cdot b$		
norme triangulaire	0	1	∇ où $a \nabla b \equiv \min(a+b,1)$	Δ où $a \Delta b \equiv \mathrm{pos}(a+b-1)$	favorise les faisceaux à évocations nombreuses

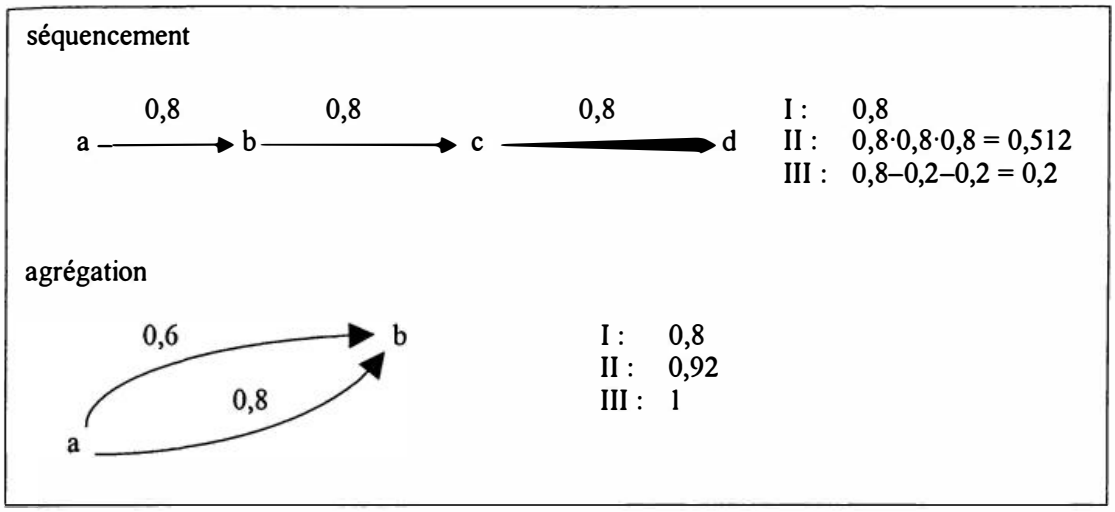


Fig. 14.1 Effets des lois de composition pour l'évocation.

Le premier favorise trop les longues chaînes d'évocation par rapport aux chaînes plus directes ; le troisième favorise exagérément les chaînes multiples même faibles[6]. Le second système est probablement le plus raisonnable.

On peut considérer comme lourde la constitution d'une table d'évocation globale à partir d'une table d'évocation immédiate. Indépendamment de l'opportunité de stratégies *matrices creuses,* on remarquera que cette table n'est à reconstruire qu'à propos de l'introduction de nouveaux mots-clés.

La reconstruction du système d'évocation viendra donc en épilogue des mises à jour du fond, et ne sera pas à la charge des consultations.

14.11 GRAPHES MULTIVALUÉS

On peut valuer des arcs par des n-uples, d'où des graphes multivalués du style : $G = \langle X, \mu : X^2 \rightarrow (A \otimes B \otimes C) \rangle$.

La définition du dioïde correspondant suppose de vérifier soigneusement les propriétés des opérateurs introduits, notamment s'ils sortent du cadre des opérations terme à terme.

[6] Affirmant ainsi qu'un faisceau de vagues associations d'idées vaut une association d'idée franche et directe.

EXEMPLE. Soit un réseau routier. Associons à chaque segment un quintuplet *(nom, longueur, vitesse max, hauteur max, poids max)*.

En séquence, les noms se concatènent, les longueurs s'ajoutent, hauteur et poids se combinent par *min*. Par contre, les vitesses se combinent indirectement, par addition des durées associées ; avec : $v_{12} = (l_1 + l_2)/(l_1/v_1 + l_2/v_2)$. Sauf pour la vitesse, les combinaisons séquentielles se feraient donc champ par champ.

On préfèrera les n-uples du type ci-contre, dont la combinaison séquentielle se fait entièrement champ par champ à l'aide des opérateurs associatifs respectifs qui nous conservent clairement un monoïde de valuation (par quintuplets).

nom	longueur	duréeMin = longueur/ vitesseMax	hauteur max	poids max
concaténation	+	+	min	min

L'agrégation est plus délicate :

- si l'on cherche le meilleur chemin sur l'ensemble des critères, une agrégation sera l'union des deux chemins, sauf si l'un des deux est préférable ou équivalent à l'autre pour chacun des critères (préférence unanime) : cette agrégation peu directive supposerait une composition séquentielle généralisée ;
- sinon, poser que l'agrégation associe un t-uple unique à toute paire de chemins de même extrémité revient à introduire dès ce niveau des arbitrages entre critères : en général, on retiendra le chemin le plus faisable (poids, hauteur, largeur...) ou, à faisabilité égale, le chemin optimal (longueur, durée...).

14.12 EXERCICES

14.12.0 Table des distances

Calculer une table des distances complète d'après la table 14.6 relative aux liaisons directes.

Tracer le graphe valué correspondant à la table initiale ; s'en servir pour vérifier votre calcul matriciel.

Tableau 14.6 Table des distances « liaisons directes ».

de à	a	b	c	d	e	f	g	h
a	0	20	∞	∞	∞	∞	∞	50
b	20	0	25	∞	∞	∞	∞	57
c	∞	25	0	18	∞	50	48	48
d	∞	∞	18	0	75	60	63	∞
e	∞	∞	∞	75	0	70	∞	∞
f	∞	∞	50	60	70	0	80	∞
g	∞	∞	48	63	∞	80	0	30
h	50	57	48	∞	∞	∞	30	0

14.12.1 Résistance d'un réseau électrique

On suppose que la table 14.6 représente maintenant les résistances d'un réseau électrique. Pour trouver les résistances globales point à point, montrer d'abord que la structure <ℝ, ∞, 0, //,+> est bien un dioïde.

14.12.2 Enigme

A l'aide d'un (multi)graphe marqué, chercher la solution la plus simple de cette énigme : « Une femme mariée à un homme qui la trompe avec la femme de son amant, laquelle trompe son mari avec le sien et qui en est réduite à tromper son amant avec celui de sa femme parce que son amant est son mari et que la femme de son époux est la maîtresse d'un homme déshonoré par l'amant d'une femme dont le mari trompe sa maîtresse avec la femme de son amant ne sait plus où elle en est ni ce qu'elle doit faire pour ne pas compliquer encore une situation qui l'est déjà suffisamment comme ça. »[7]

14.12.3 Mise en exploitation d'un gisement minier (GONDRAN & MINOUX)

La mise en exploitation d'un nouveau gisement minier demande la réalisation de diverses tâches, soumises aux contraintes d'antériorité données au tableau 14.7.

Définir une dizaine d'événements marquant l'exécution de ces tâches : début, finA, finB, ... fin.

Définir une matrice carrée à raison d'une ligne et d'une colonne par événement, chaque case (m, n) contenant la durée de la tâche à faire de m à j (resp. la durée de la plus longue des tâches à faire de m à n).

Montrer que <ℝ, –∞, 0, max, +> est un dioïde adapté à la recherche des dates au plus tôt de ces événements.

Trouver la *séquence critique*, i.e. la séquence la plus longue de *début* à *fin*.

Tracer le graphe des précédences, en commençant par la séquence critique disposée le long d'un axe horizontal.

Montrer qu'on peut trouver les dates au plus tard des mêmes événements en s'intéressant au graphe inverse.

Quels sont les événements pour lesquels on a la plus grande marge ? (i.e. la plus grande différence entre date au plus tôt et au plus tard).

On suppose maintenant que, pour chaque tâche, la durée en jours est en première approximation son coût initial ; mais que, par des allocations de moyens différentes, il est possible de modifier durée et coût en maintenant leur *produit constant* (360, 360) pouvant devenir (300, 432) par exemple.

Coût du projet en l'état ?

Coût du projet si on doit ramener sa durée à 1000 jours ?

14.12.4 Irrigation

Votre société vient de prendre la gérance d'un système de canaux d'irrigation.

Pour chaque portion de canal, la table 14.8 donne la capacité (en m^3/s). Pour trouver pour tout couple x-y le débit maximal permis par le réseau, définir le dioïde de calcul, ses opérations, ses éléments remarquables. Compléter en conséquence la table 14.8, et calculer pour tout couple x-y le débit maximal permis par le réseau.

---

[7] Pierre DAC, *Pensées*, Le Cherche-Midi, 1972.

Tableau 14.7 Mise en exploitation d'un gisement minier.

Tâche	Définition	Durée (jours)	Tâches antérieures
a	obtention du permis d'exploitation	120	aucune
b	établissement d'une piste de 6 km	180	a
c	transport et installation à pied d'œuvre de 2 sondeuses	3	b
d	construction de bâtiments provisoires pour le bureau des plans et le logement des ouvriers-sondeurs	30	b
e	goudronnage de la piste	60	b
f	adduction d'eau	60	d
g	campagne de sondage	240	c, d
h	forage et équipement de 3 puits	180	g, e, f
i	transport et installation au fond du matériel d'exploitation	30	j, h
j	construction de bureaux et logements des ouvriers et ingénieurs	240	f, e, g
k	traçage et aménagement du fond	360	j, h
l	construction d'une laverie	240	j, h

Tableau 14.8 Débits des segments de canaux.

de à	a	b	c	d	e	f	g	h
a	0	50						
b		0		42				
c	35		0		65			
d				0				26
e				83	0		18	
f					40	0		
g							0	35
h								0

Vraisemblance des données : expliquez pourquoi le graphe topologique associé doit admettre une fonction ordinale.

Se servir d'une telle fonction pour tracer un graphe du réseau. Intérêt pour la convergence des calculs ?

On suppose que des stations de pompage peuvent alimenter c à raison de 100 m^3/s et f à raison de 40 m^3/s. Définir le débit maximal d'arrivée en h, et les segments de canaux qui se trouvent saturés.

On suppose qu'on peut accroître le débit d'un segment de canal à raison de 100 000 oros par m^3/s supplémentaire. Proposer un plan d'investissement permettant d'accroître le débit en h en vue de futures extensions.

14.13 PROBLÈMES

14.13.1 Tracés définis point par point (suite du problème 13.8.1)

Dans le style des codes de Freeman, on considère l'ensemble C = {1, 2, 3, 4, 5, 6, 7, 8, 9, $\underline{1}$, $\underline{2}$, $\underline{3}$, $\underline{4}$, $\underline{5}$, $\underline{6}$, $\underline{7}$, $\underline{8}$, $\underline{9}$} des codes élémentaires de tracé ou déplacement dans un plan, où

- les chiffres soulignés correspondent aux *déplacements traçants*,
- les chiffres normaux correspondent aux *déplacements purs*,
- et le chiffre 5 dénote un *déplacement ineffectif*.

A partir de C, de l'union et de la composition on définit un dioïde C^{+*}.

Soient d et t des séquences désignant respectivement des déplacements et des tracés.

On montrera que toute séquence t_1dt_2 peut se ramener à une somme $t_1 + d_1dt_2$, où d_1 calque les déplacements de t_1.

On en déduira que toute image pixellisée peut se ramener à une *forme canonique* $\Sigma\, d_it_i$ comprenant un produit d·t par sous-image, chaque séquence d_i étant elle-même sous la forme g^mh^n, où g et h désignent deux directions contiguës de déplacement.

Pour rendre les images comparables, peut-on alors *ordonner* les d_it_i ?

Comment décider qu'une image ainsi décrite est la translatée d'une autre ?

14.13.2 Affinogrammes

On modélise les relations entre individus d'un groupe par un graphe valué $G = \langle X, \mu : X^2 \to V = \{-, 0, +\}\rangle$ dans lequel :

- il existe un arc marqué + de x vers y si x *apprécie* y,
- il existe un arc marqué – de x vers y si x *est allergique à* y,
- l'absence d'arc représente un tiers-cas d'indifférence, noté 0 si nécessaire.

Dans ce système où V = {–, 0, +}, redéfinir le concept de transitivité ; les différents cas pourront être figurés par un triangle formé d'un chemin (x, y, z) et de sa corde (x, z) ; caractériser

- les graphes de « fraternité universelle »,
- les graphes de « jalousie universelle »,
- les graphes des « alliances de nécessité » (les ennemis de mes ennemis sont mes amis)

Peut-on bâtir des classes d'affinité ?

Y a-t-il des conditions particulières pour obtenir un système équilibré, i.e. respectant le principe : « aucune flèche négative entre sommets d'une même classe » ?

Réinterpréter les fermetures transitives.

14.14 PISTES DE RÉFLEXION

14.14.1 Relations renseignées (Jean KUNTZMANN, 1972)

On considère que tout graphe valué introduit un nouveau style de relations, dites relations *renseignées*.

Soit A l'ensemble des valeurs ou renseignements. A quelle condition sur A peut-on dire qu'à chaque relation renseignée $R_v = (X, A, \mu : X^2 \to A)$ correspond une relation booléenne $R_B = (X, A, \mu_B : X^2 \to \mathbb{B})$, qu'elle généralise ?

Que faut-il à *l'ensemble de renseignement* A pour permettre la combinaison de relations ? pour pouvoir dire que les relations $R_v = (X, A, \mu : X^2 \longrightarrow A)$ sont en fait l'expression de relations dans une logique non standard dont A serait l'ensemble de vérité ?

Que deviennent les fermetures transitives ? quelle est maintenant leur signification ? sont-elles toujours calculables en un temps fini ?

Ex : prendre A = { 00 = impossible, 01 = vrai en général mais faux ici, 10 = vrai ici mais faux en général, 11 = toujours vrai } (logique de Reischer et Davio *in* Thaysse).

14.14.2 Physique qualitative

On souhaite formaliser les raisonnements intuitifs en physique, guidant des actions pratiques telles que *jeter une boule de papier dans une corbeille.*

Pour cela, on modélise les relations entre variables d'état d'un système par un graphe valué

$G = \langle X, \mu : X^2 \to V = \{-, 0, +\}\rangle$, tout arc (x, y) étant valué –/ 0 /+ selon que $\partial y / \partial x$ est (suffisamment) négatif, (négligeable ou) nul, (suffisamment) positif.

Montrer comment attribuer un signe aux chemins d'influence. Discuter la signification des circuits positifs ou négatifs en termes de stabilité du système.

Comment agréger 2 influences indirectes disjointes de x sur y ? lorsqu'elles sont de même signe ? de signes distincts ?

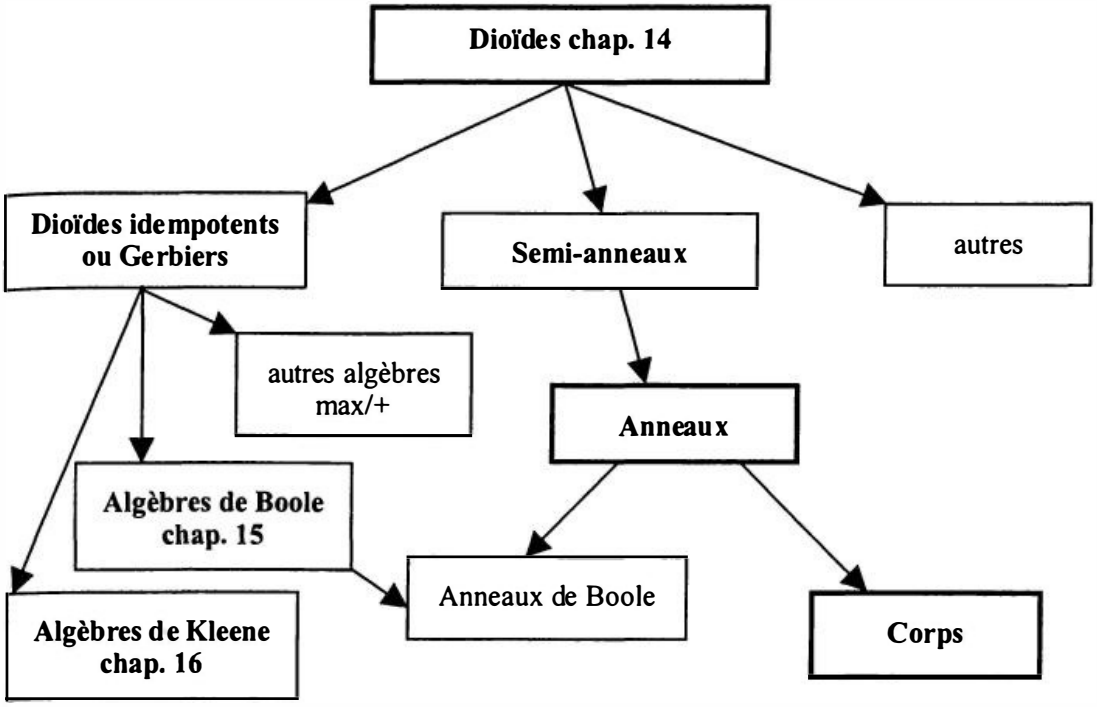
Comment échapper au quantitatif pour juger de l'effet global d'influences conflictuelles ? Un formalisme qui exploiterait une idée de force, voire d'ordre de grandeur des $\partial y / \partial x$ conviendrait-il ?

14.14.3 1001 algorithmes sur les graphes

Afin de disposer à peu de frais de 1001 algorithmes sur les graphes, réaliser une bibliothèque générique de traitement matriciel style Dioïde dans le langage objet de votre choix (conseil : étudier d'abord Gondran & Minoux, chap. I à III).

14.15 FAMILLE DES DIOÏDES

... classiques et/ou propres aux mathématiques discrètes.



14.16 LECTURES

BERGE C., *Théorie des graphes et ses applications,* Dunod, 1958, 2e édition, 1963.

FAURE R., LEMAIRE J., *Mathématiques pour l'informaticien,* tome I, chap. VI, Coll. Programmation, Ed. Gauthier-Villars, 1973.

GONDRAN & MINOUX, *Graphes et Algorithmes,* chap. III, collection EDF n° 37, Eyrolles, 1979-85-95...

KAUFMANN A., BOULAYE G., *Théorie des treillis en vue des applications,* chap. I, Ed. Masson, 1978.

KAUFMANN A., PICHAT E., *Méthodes mathématiques non numériques et leurs algorithmes,* vol. 2. Algorithmes de recherche de chemins et problèmes associés Ed. Masson, 1977.

KUNTZMANN J., *Théorie des réseaux*, Dunod, 1972.

LAURIÈRE J.C., *Intelligence artificielle* : tome 1 : Résolution de Problèmes, Eyrolles.

MACOWICZ M., *Approche générique du traitement des graphes,* thèse INSA Lyon, 7 nov. 1997, 160 p. environ.

PAIR C., DERNIAME J.C., *Problèmes de cheminement dans les graphes,* Monographie Afcet, Dunod, 1972.

PETERSON W.W., WELDON E.J. Jr, *Error Correcting Codes*, MIT Press, 1972.

ROY B., *Algèbre moderne et théorie des graphes*, Dunod, tome 2, 1970.

coll., *Petite encyclopédie des mathématiques,* chap. XVI, Edition Pagoulatos, Didier distr.

CHAPITRE 15

ALGÈBRE DE BOOLE

15.1 INTÉRÊT

L'algèbre de Boole est née en 1854 de la volonté de George Boole de ramener les problèmes de logique classique (ou scolastique), alors traités par des raisonnements, parfois des diagrammes, à des calculs algébriques assimilant 0 à faux et 1 à vrai, de telle façon que les raisonnements (tels que présentés chapitre 0) soient remplacés par l'évaluation d'une formule logique, ou, au pire, la résolution d'un système d'équations logiques. Ce but amena G. Boole à un formalisme distinct de l'algèbre traditionnelle, avec notamment l'association à la conjonction d'un produit idempotent (i.e. tel que $x \cdot x = x$) .

En 1938, Shannon remarque que cette algèbre de Boole permet de modéliser le fonctionnement des circuits à relais, utilisés en téléphonie et en automatisme.

A la même époque, en Allemagne, Konrad Zuse utilise pour ses prototypes de calculateurs l'arithmétique binaire[1] car le calcul propositionnel lui permet une définition logique des composants nécessaires, qu'il réalise d'abord de façon purement mécanique, puis à l'aide de relais[2]. En 1941, son Z3 est le premier calculateur programmable opérationnel, avec une unité virgule flottante à 1+7+14 bits, une mémoire à relais de 64 mots, et un programme externe sur ruban de papier.

L'informatique actuelle conjugue deux évolutions :

- le *traitement d'informations de plus en plus variées*, généralement discrètes mais pas forcément numériques, qu'en dernière analyse le matériel traite par manipulation logico-arithmétique de codes binaires ;
- des *technologies de plus en plus efficaces,* glissant des relais aux triodes dans les années 40, des triodes aux transistors à partir de 1960, puis aux circuits intégrés SSI, MSI, LSI, VLSI...[3] dont la densité d'intégration décuple tous les 4 à 5 ans.

Cependant, l'algèbre de Boole permet d'expliquer dans différents styles les bases des réseaux logiques de contacts, des circuits électroniques binaires et d'autres dispositifs de décision binaire sans avoir à décider si c'est pour l'informatique ou l'automatique...

---

1 Déjà abordée au XVIIe siècle par Leibniz, et que K. Zuse étend à la virgule flottante, définie en 1914 par Torres y Quevedo.

2 En France on trouve à cette époque des propositions similaires dans la thèse de Louis Couffignal.

3 *(Small / Medium / Large / Very Large) Scale Integration.*

15.2 ALGÈBRE DE BOOLE ÉLÉMENTAIRE

On part d'un ensemble à deux valeurs $\mathbb{B} = \{0, 1\}$, représentant faux et vrai, qu'on munit d'opérations + (pour ***ou***) et · (pour ***et***) définies par les tables suivantes.

Tableau 15.1 Opérateurs + et · dans $\mathbb{B}$.

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

Soient a, b, c... des variables à valeur dans $\mathbb{B}$. Au pire par énumération des cas, on établit les propriétés suivantes.

Tableau 15.2 Propriétés des opérateurs + et · dans $\mathbb{B}$.

propriétés	$<\mathbb{B}, 0, 1, +>$	$<\mathbb{B}, 1, 0, \cdot>$
associativité	$(a + b) + c = a + (b + c)$	$(a\cdot b)\cdot c = a\cdot(b\cdot c)$
élément neutre	$a + 0 = a = 0 + a$	$a\cdot 1 = a = 1\cdot a$
élément absorbant	$a + 1 = 1 = 1 + a$	$a\cdot 0 = 0 = 0\cdot a$
idempotence	$a + a = a$	$a\cdot a = a$
commutativité	$a + b = b + a$	$a\cdot b = b\cdot a$
	$<\mathbb{B}, 0, 1, +>$ monoïde idempotent et commutatif	$<\mathbb{B}, 1, 0, \cdot>$ monoïde idempotent et commutatif
ordre préfixe	$a \leq a + b$	$a \geq a\cdot b$
réflexif	$a \leq a$	$a \geq a$
transitif	$a \leq b, b \leq c \Rightarrow a \leq c$	$a \geq b, b \geq c \Rightarrow a \geq c$
antisymétrique	$a \leq b, b \leq a \Rightarrow a = b$	$a \geq b, b \geq a \Rightarrow a = b$
treillis	$\sup(a,b) = a + b$	$\inf(a,b) = a\cdot b$

$<\mathbb{B}, \omega = 0, \varepsilon = 1, +, \cdot>$ est un dioïde « riche » dont les 2 opérations possèdent de plus des propriétés mutuelles indiquées dans le tableau 15.3.

Tableau 15.3 Propriétés mutuelles des opérateurs + et · dans $\mathbb{B}$.

propriétés +/·	$<\mathbb{B}, \omega = 0, \varepsilon = 1, +, \cdot>$	
distributivité	$a\cdot(b + c) = a\cdot b + a\cdot c = b\cdot a + c\cdot a = (b + c)\cdot a$	$a + b\cdot c = (a + b)\cdot(a + c) = (b + a)\cdot(c + a) = b\cdot c + a$
absorption	$a + a\cdot b = a$	$a\cdot(a + b) = a$

Cette structure fonde les calculs matriciels portant sur l'existence de chemins, et sur les fermetures transitives des graphes topologiques et relations associées.

Au-delà, on introduit maintenant une troisième opération, unaire, dite ***non***, notée « ′ » et telle que :

$$0' = 1 \qquad \text{et} \qquad 1' = 0.$$

Tableau 15.4 Propriétés du « non » dans $\mathbb{B}$.

propriétés	$<\mathbb{B}, 0, 1, ', +, \cdot>$	
involution	$a'' = a$	
complémentarité	$a + a' = 1$	$a \cdot a' = 0$
dualité	$(a + b)' = a' \cdot b'$	$(a \cdot b)' = a' + b'$
automorphisme	non : $<\mathbb{B}, 0, 1, ', +, \cdot> \leftrightarrow <\mathbb{B}, 1, 0, ', \cdot, +>$	
antitonie	$a \leq b \leftrightarrow a' \geq b'$	$a \geq b \leftrightarrow a' \leq b'$

Nous appelons ***algèbre de Boole***, en bref AdB, la structure

$$< \mathbb{B}, 0, 1, ', +, \cdot, \leq >$$

ainsi formée.

15.3 MODÉLISATION BOOLÉENNE DE TECHNOLOGIES

15.3.1 Réseaux de contacts

A la base, on considère une source d'énergie et un dispositif à commander, liés par un contact laissant ou non passer l'énergie dans le dispositif, en fonction d'une grandeur extérieure qui commande le contact et par là le dispositif.

L'ensemble constitue un quadripôle : dipôle de commande (ou d'entrée) et dipôle commandé (ou de sortie) sont traversés par des courants couplés mais séparés. La sortie commandée peut être modélisée par une arête unissant les deux points de connexion du contact, marquée par l'*expression logique de la commande* rendant ce contact passant.

Un ***réseau de contacts*** est alors modélisé par un graphe symétrique, à raison

- d'un nœud par point de connexion,
- d'une arête marquée par contact,

d'une arête non marquée par connexion supplémentaire.

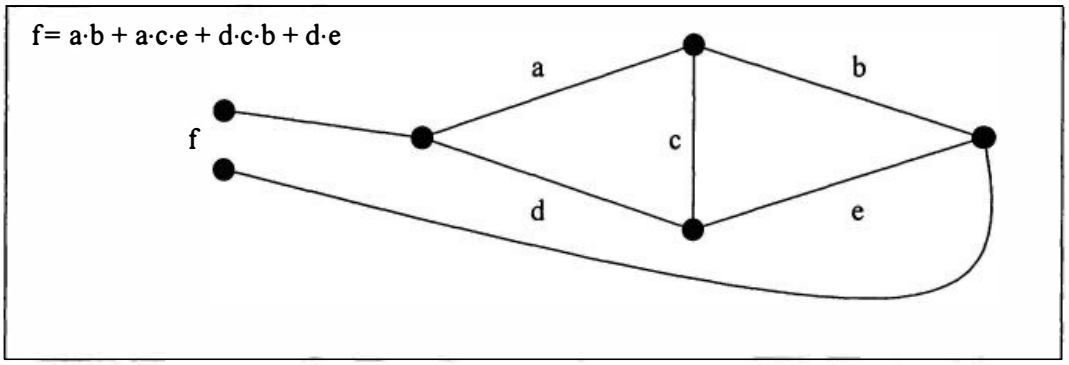


Fig. 15.1 Réseau de contacts dipolaires « en pont » vu comme un dipôle.

L'état d'un réseau de contacts se calcule en fonction des commandes connues :

- les expressions attachées aux contacts donnent soit 1 pour les contacts passants, soit 0 pour les contacts ouverts, soit une expression plus ou moins réduite si les commandes ne sont que partiellement connues,
- on combine par ET/· les valeurs ou expressions marquant 2 arêtes ou chaînes consécutives,
- on combine par OU/+ les valeurs ou expressions marquant 2 arêtes ou chaînes de mêmes extrémités.

1. On a pu ainsi réaliser avec des *relais électromagnétiques* des systèmes complexes (commutateurs téléphoniques, commandes d'ascenseurs, calculateurs, automatismes divers) ; en effet :
   - un relais comporte un bobinage de commande qui, via un noyau magnétique, soumet l'état passant / ouvert d'un contact commandé à la présence / absence d'un courant de commande dans le bobinage ;
   - le courant d'un bobinage de relais peut être commandé par une source, via un interrupteur, un bouton-poussoir... voire un réseau de contacts ;
   - le contact commandé peut autoriser/couper dans le réseau commandé un courant *bien supérieur* au courant de commande, d'où un effet amplificateur, les deux circuits restant isolés l'un de l'autre ;
   - un même bobinage peut commander simultanément plusieurs contacts, dont l'expression caractéristique sera soit l'expression de commande soit son complément selon que le contact est à ouverture ou à fermeture ;
   - le fonctionnement électromécanique n'étant pas instantané, on obtient des *effets de mémoire* si l'alimentation du bobinage M dépend d'un contact m qu'il commande, éventuellement des fonctionnements périodiques oscillants si M dépend d'un contact complémentaire m'.
2. Cette notion de réseau de contacts peut s'adapter à d'autres technologies :
   - D. Chorafas la prônait pour les *cryotrons*[4].
   - On pourrait l'adapter aux transistors (et circuits intégrés qui en découlent) en utilisant des outils booléens tels que la ***fonction duplex*** $U(x,y,z) \equiv x'y + xz$ et la ***fonction aiguillage*** $aig(e, c \rightarrow s0, s1) \equiv (s0 = ec', s1 = ec)$, ou pour certains montages en réseau une fonction $aig2(a, b, c \rightarrow s0, s1)) \equiv \{s0 = (a+b)c, s1 = (a+b)c'\}$.

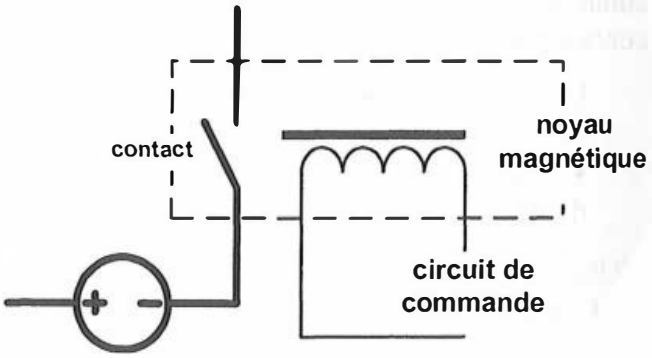


Fig. 15.2 Principe du relais électro-magnétique.

4 Dispositif constitué par deux éléments supraconducteurs : une bobine de commande et un autre circuit qui perd sa supraconductivité quand il est soumis au champ magnétique créé par la bobine (cf. effet Josephson). Utilisé comme relais (extrêmement peu consommateur d'énergie), amplificateur, mémoire. (cf. www.sciences-en-ligne.com).

15.3.2 Electronique digitale usuelle

Pour les circuits logiques électroniques à tubes (de 1938 à 1965) et à transistors (depuis 1960), le modèle de base est un *tripôle* : l'entrée commande la sortie (les tubes radioélectriques en tant qu'amplificateurs de tension, les transistors en tant qu'amplificateurs de courants), mais *entrée et sortie ont une référence commune* et doivent respecter des règles de polarité, ce qui interdit notamment les « réseaux en pont » (fig. 15.3).

Le passage d'un composant à un réseau de composants, que ce soit d'un composant à un circuit intégré ou d'un circuit intégré à un ensemble plus vaste, se modélise par des schémas fonctionnels respectant les directivités des courants, toutes les opérations logiques étant explicites (pour plus de détails sur le niveau élémentaire, voir §15.5.6 et 15.6.6). Par suite, l'évaluation de l'état d'une connexion logique se fait par composition fonctionnelle classique des entrées, constantes 0/1, variables supposées autonomes ou expressions booléennes.

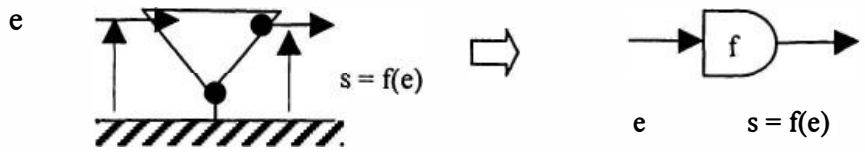


Fig. 15.3 Abstraction d'un tripôle actif en élément fonctionnel.

15.4 FONCTIONS BOOLÉENNES

Les fonctions booléennes $f : \mathbb{B}^n \to \mathbb{B}$ ont été étudiées tantôt en elles-mêmes tantôt comme nouveaux opérateurs pouvant faciliter la synthèse de fonctions plus importantes.

PROPOSITION 15.1 – Il existe exactement 2^{2^n} fonctions booléennes de n variables.

PREUVE. $\mathbb{B}^n$ comprend 2^n n-uples, et chaque fonction possible associe à chaque n-uple une valeur parmi 2. ■

EXEMPLE. Considérons le cas des *16 fonctions à 2 variables*. Nous pouvons les définir par la *table de vérité* 15.5, où la colonne centrale :

- en tête, énumère tous les couples possibles en entrée, à raison d'un couple par verticale ;
- définit chaque fonction f par une ligne contenant son *vecteur de vérité*, avec chaque valeur f(t) en regard du couple t défini sur la même verticale.

Parmi ces 16 fonctions, 2 sont en fait constantes (fonctions à 0 variable) et 4 reproduisent une entrée, en la complémentant éventuellement (fonctions à 1 variable).

Il ne reste donc que 10 fonctions intrinsèquement à 2 variables :

- le *et*, le *ou*, leurs compléments *ni* et *on* 4
- le *ou exclusif* et son complément 2
- et 4 « diverses » 4

Tableau 15.5 Les 16 fonctions booléennes de 2 variables.

x y fonction n°	0011 0101 vecteur	expression
0	0000	0
1	0001	$x \cdot y$ /et
2	0010	$x \cdot y'$
3	0011	x
4	0100	$x'y$
5	0101	y
6	0110	$x \oplus y$ /ouExclusif
7	0111	$x + y$ /ou
8	1000	$x \uparrow y = ni(x, y)$
9	1001	$x \odot y = (x \oplus y)'$
10	1010	y'
11	1011	$x' \downarrow y$
12	1100	x'
13	1101	$x \downarrow y'$
14	1110	$x \downarrow y = on(x, y)$
15	1111	1

15.5 GÉOMÉTRIE BOOLÉENNE

15.5.1 Représentation des $\mathbb{B}^n$

On peut se représenter

$\mathbb{B}$ par un *segment* de droite ou une arête,
$\mathbb{B}^2$ par un *carré* (ou cycle de taille 4),
$\mathbb{B}^3$ par un *cube*,
$\mathbb{B}^4$ par une paire de cubes convenablement couplés.

Chaque sommet de ces figures représente un n-uple de l'espace correspondant.

15.5.2 Distance de Hamming

DÉFINITION 15.1 – On prend comme distance entre 2 n-uples ou points p et q de $\mathbb{B}^n$, le nombre de composantes par lesquelles ils diffèrent. Ainsi, on a toujours :

- $d(p, q) = 0 \leftrightarrow p = q$
- $d(p, q) = d(q, p)$ /symétrie
- $d(p, r) \leq d(p, q) + d(q, r)$ / inégalité du triangle

Cette distance[5] apparaît comme la taille de la chaîne la plus simple allant dans $\mathbb{B}^n$ d'un point à l'autre.

[5] Introduite pour les besoins d'une théorie des codes détecteurs et correcteurs d'erreur (cf. aussi W.W. PETERSON, op. cit. chap. 14).

15.5.3 Représentation géométrique des fonctions booléennes

A chaque variable est associée une dimension.

Chaque sommet de la représentation de $\mathbb{B}^n$ est marqué ssi la fonction vaut 1 pour le tuple correspondant : *la fonction est assimilée à l'ensemble de ses 1*.

A deux sommets adjacents on associe le segment qui les sépare.

A quatre sommets sur un cycle de taille 4 on associe la face que ce cycle délimite...

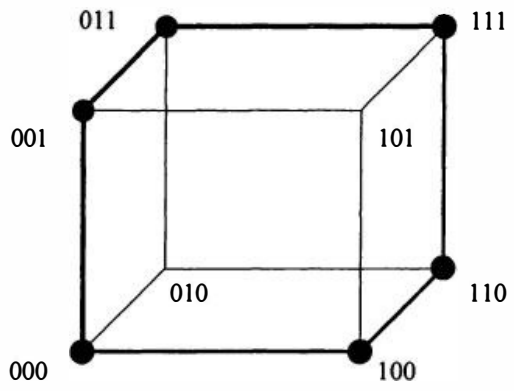


Fig. 15.4 Fonction $\mathbb{B}^3 \to \mathbb{B}$ dite « ceinture du cube ».

15.5.4 Points & monômes canoniques

Considérons une fonction de n variables valant 1 pour un n-uple et un seul. Il lui correspond exactement un point de $\mathbb{B}^n$.

PROPOSITION 15.2 – Chaque n-uple ou point de $\mathbb{B}^n$ peut être décrit par un monôme à n variables dit *monôme canonique*, chaque variable étant présente

- sous forme directe, si la composante correspondante du n-uple vaut 1,
- sous forme complémentée, si la composante correspondante du n-uple vaut 0.

EXEMPLE. Seront considérées équivalentes les désignations f(a, b, c, d, e) = {00101} et f(a, b, c, d, e) = a′b′cd′e.

15.5.5 Première forme canonique

PROPOSITION 15.3 – *Toute fonction booléenne peut être définie par une somme de monômes canoniques.*

PREUVE. La bijection *point* ↔ *monôme canonique* s'étend pour les fonctions quelconques à *collection de points* ↔ *somme de monômes canoniques*. ■

On peut imposer que les monômes des sommes obtenues suivent un ordre d'énumération fixe.

EXEMPLE. Soit f(a, b, c) = {000, 001, 101, 111, 110, 010} /de proche en proche

= a′b′c′+a′b′c+ab′c+abc+abc′+a′bc′ /première forme canonique

= a′b′c′+a′b′c+a′bc′+ab′c+abc′+abc, /réécrite dans l'ordre {0, 1, 2, 5, 6,7} des transcriptions décimales des n-uples.

15.5.6 Sommes de produits

En fait, les points d'une fonction sont souvent combinables par 2, 4, 8... correspondant à des monômes réduits à n – 1, n – 2, n – 3... variables.

Outre la première forme canonique, il existe de nombreuses façons d'écrire une fonction en somme de produits, menant à des réalisations de coût assez variable.

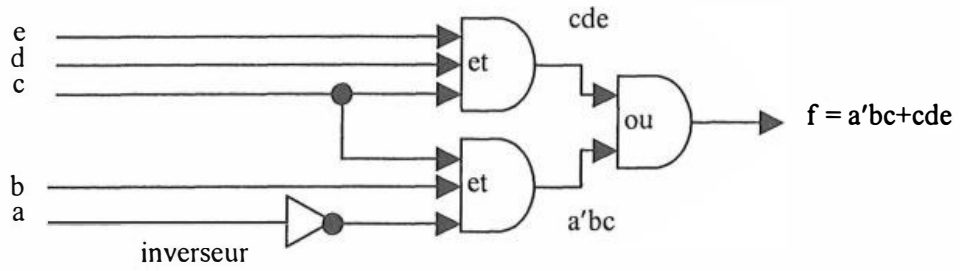


Fig. 15.5 Réalisation d'une fonction booléenne en somme de produits.

Si l'on ne disposait strictement que d'opérateurs à 2 variables, il faudrait k – 1 opérateurs ET pour un monôme à k variables, et m – 1 opérateurs OU pour faire la somme de m monômes. En première approximation, le coût d'une somme de produits est ainsi donné par le nombre total d'occurrences de variables, les écritures les moins coûteuses étant les sommes regroupant peu de monômes de peu de variables.

Le problème est ouvert, car pour un monôme donné d'un système à n variables, chaque variable peut soit y figurer sous forme directe ou complémentée, soit ne pas y figurer.

Pour n variables il y a donc $3^n - 1$ monômes quelconques, dont 2^n monômes canoniques. Cette abondance favorise les redondances, et une réalisation efficiente supposera de trouver la plus petite collection de monômes les plus simples réalisant exactement la fonction.

EXEMPLE. Soit f(a, b, c) = {000, 001, 101, 111, 110, 010} /de proche en proche
= a'b'c' + a'b'c + ab'c + abc + abc' + a'bc' /première forme canonique

Une réalisation calquée sur cette forme suppose 2 ET à 2 entrées par produit, et 1 OU à 2 entrées par somme, soit ici 12 ET à 2 entrées et 5 OU à 2 entrées, soit 17 *outils logiques*, à comparer aux 18 lettres employées.

Si nous nous inspirons des proximités, et codons des segments plutôt que des points, à {000, 001} correspond a'b', à {101, 111} correspond ac, à {110, 010} correspond bc', et on peut écrire f(a, b, c) = a'b' + ac + bc'. Cette forme, qui n'est pas canonique, coûte en tout 5 outils à 2 entrées, et n'emploie en tout que 6 lettres : elle mène à une réalisation environ 3 fois moins coûteuse que la définition canonique !

15.5.7 Diagramme de Veitch-Karnaugh

Déjà vers 1300, dans son *Ars Magna,* Raymond Lulle utilisa un diagramme figurant les rapports logiques qu'il avait établi entre dogmes principaux et dogmes secondaires.

Charles DODGSON[6] en a aussi utilisé dans sa logique sous une forme variable.

---

[6] Dit Lewis CAROLL (cf. chap. 0).

Veitch a défini plus récemment un diagramme systématique rendu opérationnel par Karnaugh, et utilisable de 2 à 6 variables.

On part d'un carré. On alloue une variable par côté, en tournant autour du carré : la première étant au nord, la seconde à l'ouest, la troisième au sud, la quatrième à l'est, la cinquième de nouveau au nord... Pour chaque variable, on divise en deux le carré (resp. les carrés, les rangées) qui lui font face. Pour une même direction, les variables sont alternées comme dans un code de Gray : a′ / a subdivisé c′/c deviendra a′c′/ a′c/ ac / ac′ puis a′c′e′/a′c′e/a′ce/a′ce′/ace′/ace/ac′e/ac′e′ *de façon à ce qu'une ligne corresponde à un changement de variable et un seul.*

Une fonction se représente à raison d'une case par monôme canonique, une barre à 2 cases pour un monôme à n – 1 variables, un carré ou une barre à 4 cases pour un monôme à n – 2 variables...

EXEMPLE. Soit f(a, b, c) = {000, 001, 101, 111, 110, 010} = a′b′c′ + a′b′c + ab′c + abc + abc′ + a′bc′.

Elle est représentée par la figure 15.6. On voit apparaître la forme réduite f(a, b, c) = a′b′ + ac + bc′ si on regroupe les cases « séparées par un axe », i.e. différant par une variable et une seule ... les bords pouvant tenir lieu d'axes (comme si le diagramme était tracé sur un cylindre ou une sphère).

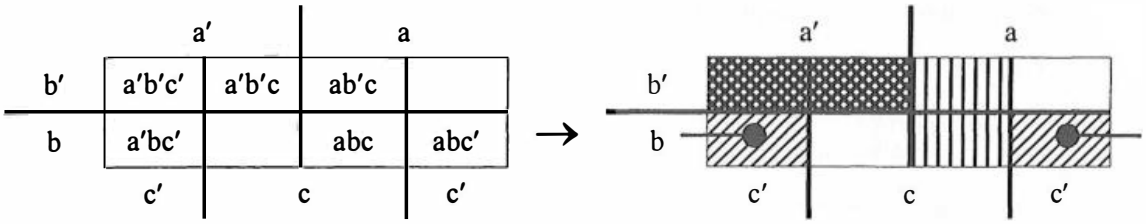


Fig. 15.6 Diagramme de Veitch-Karnaugh d'une fonction booléenne à 3 variables.

15.6 BASES D'UNE FONCTION

15.6.1 Monômes premiers

REMARQUE. En algèbre de Boole, on a toujours a·b ≤ a : tout produit est inférieur à ses facteurs.

DÉFINITION 15.2 – On appellera ***facteur strict*** d'un monôme m un monôme μ formé à partir de m en ôtant au moins une variable.

Ainsi m = μ·p, où p regroupe les variables de m non retenues dans μ, d'où m ≤ μ par principe, et m < μ dès qu'au moins une lettre a été effectivement retirée.

EXEMPLE. Les facteurs stricts de a′bc sont bc (p = a′), a′c (p = b) et a′b (p = c), puis a′, b et c.

DÉFINITION 15.3 – Considérons une fonction écrite en somme de produits, avec f = m + S, où m désigne un monôme isolé et S la somme des autres monômes. Alors m ≤ f (ou m·f = m). On dira m ***monôme premier*** *pour f* s'il n'existe pas de monôme μ facteur strict de m tel que μ ≤ f (ou μ·f = μ ou f = μ + S).

EXEMPLE. Soit h = abc′ + ab′c′ + def.

- dans cette écriture, *def est monôme premier de h* ;

- ni abc′ ni ab′c′ ne le sont, mais leur facteur commun ac′ l'est :

soit h = abc′+S, avec S = ab′c′+def ; alors

ac′+S	=	ac′+ab′c′+def	par définition
	=	(abc′+ab′c′)+ab′c′+def	en développant ac′
	=	abc′+(ab′c′+ab′c′)+def	par associativité
	=	abc′+ab′c′+def	par idempotence
	=	h	

tandis que a + S > h et c′+ S > h.

→ *ac′ est monôme premier de h, comme plus grand facteur de m restant inférieur à h.*

15.6.2 Bases premières

DÉFINITION 15.4 – On appelle ***base première*** d'une fonction f, son écriture en une somme de monômes premiers.

PROPOSITION 15.4 – Une telle base existe toujours.

PREUVE. Supposons une fonction à n variables sous forme canonique. Chacun de ses monômes canoniques m possède 2^n facteurs (y compris 1 et lui-même), et l'un au moins, μ, est premier : par définition, remplacer m par μ ne change rien à la fonction. Ainsi passe-t-on de la forme canonique à une base première, en remplaçant successivement chaque monôme canonique par l'un ou l'autre de ses facteurs monôme premier. ■

EXEMPLE. Soit f = {000, 001, 011, 111, 110, 100}. (cf. 15.5.3). Nous pouvons écrire :

f(a, b, c)	= a′b′c′ + a′b′c + a′bc + abc + abc′ + ab′c′	/ première forme canonique
	= a′b′ + a′c + bc + ab + ac′ + b′c′	/ base première

Une base première peut être redondante, mais *ne comprend que les monômes* m *les plus simples possibles compatibles avec la fonction f donnée,* i.e. tels que m·f = m.

Pour une synthèse optimale d'une fonction f, il nous faut :

- construire la ***base première complète*** de f, formée de tous les monômes premiers de f et d'eux seuls ;
- en confrontant cette base à la liste des points de f, en tirer des *bases premières irredondantes*, formées de listes nécessaires et suffisantes de monômes premiers ;
- en s'inspirant si nécessaire d'aspects technologiques, arrêter les bases de coût minimales parmi les bases irredondantes.

15.6.3 Méthode des consensus

DÉFINITION 15.5 – Soient deux monômes am et a′p comportant une variable et une seule sous forme directe dans l'un, sous forme complémentée dans l'autre. On a :

am = am + amp et a′p = a′p + a′mp par absorption,

am + a′p	= am + a′p + amp + a′mp	
	= am + a′p + (a+a′) mp	/par distributivité
	= am + a′p + *mp*	/car a+a′ = 1

mp sera dit ***consensus*** [7] de am et a′p, et noté mp = consensus(am, a′p).

---

[7] Quine, Tison.

Dans tous les autres cas, on pose que le consensus de deux monômes est nul.

Supposons la fonction dont nous cherchons la base première, définie par une liste de monômes L.

En supposant cette liste L *dynamique,* nous la transformons en base première (complète) en appliquant le procédé ci-après.

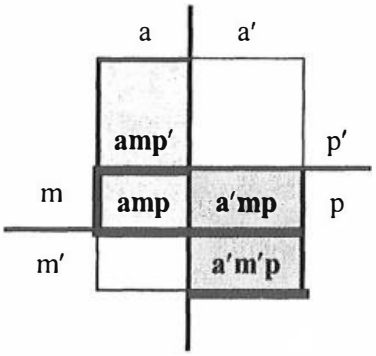


Fig. 15.7 Le consensus *mp* comme terme transversal à *am* et *a′p*.

action méthodeDesConsensus(> L : liste dynamique de monômes >)

pour chaque monôme m du premier à l'avant-dernier de L

pour chaque monôme p, du successeur de m au dernier de L

ajouter(consensus (m,p)) à (L) après (rang(m))

fin méthodeDesConsensus

action ajouter(> c :monôme) à (>L : liste dynamique de monômes >) après (> k :rang)

```
entier j ←k ;
tantque j ≤ longueur(L)          -- à réévaluer à chaque boucle
    si c< L[j], retour ;         -- c ineffectif (nul ou inférieur à un monôme
                                 -- connu)
    si c ≥ L[j], supprimer L[j]  -- contraction (répétable)
    sinon incrémenter j          -- sinon avance[8]
L ← L · {c}                      -- expansion (1 poste) si c n'était inférieur ou
                                 -- égal à aucun monôme de L
```

fin ajouter()à()après()

EXEMPLE. Soit une fonction f = {00x0, 00x1, 011x, 111x} comprenant 4 arêtes disjointes, donc 8 points. Quelle est sa base première complète ?

En appelant a, b, c, d les variables, nous avons successivement,

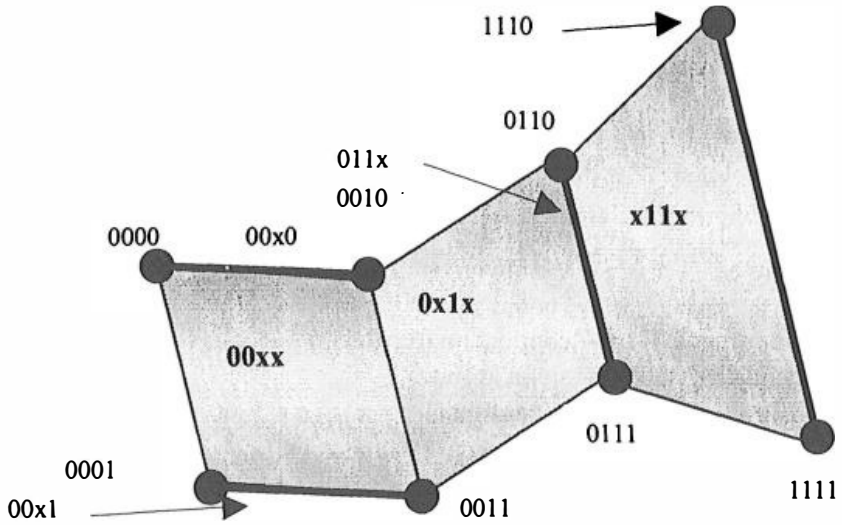
f = a′b′d′ + a′b′d + a′bc + abc
= f+ consensus(a′b′d′, a′b′d)
= f+ *a′b′*
= a′bc + abc + a′b′ / a′b′ absorbe a′b′d′ et a′b′d
= a′bc + abc +a′b′ + consensus (a′bc + abc)
= a′bc + abc + a′b′ + *bc*
= a′b′ + bc / bc absorbe a′bc et abc
= a′b′ + bc + consensus (a′b′ + bc)
= a′b′ + bc + *a′c* / forme finale : aucun autre consensus n'est produit.

[8] Rappel : 2 monômes booléens c et L(j) ne sont pas forcément comparables.

Ce qui correspond à l'exécution simulée ci-après, où les noms de variables restent implicites, et où l'on voit le codage ternaire des monômes partiellement ou entièrement réduits (les marques 0, 1, x correspondant respectivement à une variable complémentée, directe, ou réduite).

Tableau 15.6 Une application de la méthode des consensus.

liste	ajouter	liste	ajouter	liste	ajouter	liste	ajouter	liste	ajouter
00x0		/////////							
00x1	00xx	/////////							
011x		011x		/////////					
111x		111x	x11x	/////////					
	>>>	00xx		00xx		00xx		00xx	
			>>>	x11x	0x1x	x11x		x11x	
					>>>	0x1x	ϕ	0x1x	ϕ



en gras : données (segments) → monômes premiers (faces obtenues)
(00xx, x11x obligatoires, 0x1x redondant)

Fig. 15.8 Représentation graphique du calcul des consensus.

La fonction f peut donc déjà être réalisée par :

- une somme de 8 monômes à 4 variables, sa *première forme canonique*, de coût 32,
- une somme de 4 monômes à 3 variables (parmi 4) : *données* de l'exemple, de coût 12,
- une somme de 3 monômes à 2 variables (parmi 4) : *base première complète*, de coût 6.

15.6.4 Bases irredondantes

La base première complète associe à chaque point de la fonction le(s) plus gros monôme(s) le réalisant, sans dépasser la fonction, les plus gros monômes étant ici les

moins chers. Mais cette base première peut comporter des redondances : on s'en assure par une méthode de couverture (cf. sect. 1.7), mettant en regard

- côté *objectifs,* chaque point de la fonction à réaliser,
- côté *moyens* éventuellement redondants, chaque monôme premier.

DÉFINITION 15.6 – Une *base première* est dite ***base première irredondante*** si elle réalise la fonction f cherchée, mais que ce n'est plus vrai quand on lui retire l'un de ses monômes.

PROPOSITION 15.5 – Une base première B d'une fonction f est irredondante s'il n'existe pas de base première de f plus fine que B.

DÉFINITION 15.7 – Un monôme d'une *base première* est dit ***monôme obligatoire*** s'il est seul à couvrir un point de la fonction.

PROPOSITION 15.6 – Soit B une base première pour une fonction f. Toute base irredondante pour f plus fine que B contient l'ensemble des monômes de B obligatoires pour f.

PREUVE. Soit m un monôme obligatoire d'une base première B, seul à couvrir un point p de f. Si m n'est pas retenu, p n'est pas couvert et la base formée ne couvre plus f. Donc toute base formée à partir de B pour réaliser f doit contenir chaque monôme de B obligatoire pour f. ■

DÉFINITION 15.8 – Un monôme non obligatoire d'une *base première B(f)* est dit ***monôme utile*** s'il couvre au moins un point de f non couvert par l'ensemble des monômes obligatoires, ***monôme inutile*** sinon.

PROPOSITION 15.7 – Soit B une base première pour une fonction f. Une base irredondante pour f plus fine que B ne contient aucun monôme inutile de B(f).

PREUVE. Par l'absurde. Soit m un monôme inutile de B(f), et soit $B^{\sim}(f)$ une couverture irredondante de f plus fine que B, contenant m. Alors, m ne couvrant aucun point en propre, $B^{\sim}(f)-m$ est une couverture de f encore plus fine que $B^{\sim}(f)$, qui ne peut donc être irredondante. ■

Formation d'une base première irredondante d'une fonction f à partir d'une base première B

On détermine d'abord la collection B_0 des monômes premiers de B obligatoires pour f.

- Si B_0 couvre tout f, c'est terminé.

- Si B_0 couvre seulement une partie g de f, on réduit B à une base première réduite $B_r = B - B_0 - B_1$, formée en retirant de B la collection B_0 des monômes obligatoires de f qui couvrent g, *et* la collection B_1 des monômes inutiles, monômes non obligatoires qui ne couvrent rien de f – g. On procède alors à la couverture des points de f – g par les monômes de B_r. Ce *problème résiduel* est souvent traité de façon arborescente.

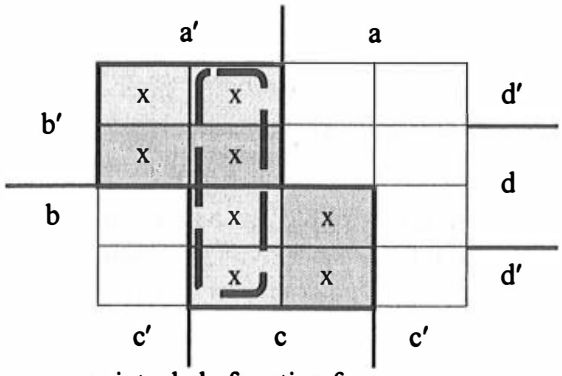
Dans le cas ci-dessus, fig. 15.8, 00xx et x11x sont obligatoires, et leur somme couvre la totalité des points de f. Le monôme 0x1x est donc totalement inutile.

D'où l'unique écriture irredondante (et, de ce fait, minimale) :

$$f = \{00xx, x11x\} = a'b' + bc, \textit{ de coût 4}$$

Tableau 15.7 Couverture des points d'une fonction.

f	00x0		00x1		011x		111x	
base	0000	0010	0001	0011	0110	0111	1110	1111
00xx	x	x	x	x				
0x1x		x		x	x	x		
x11x					x	x	x	x
	1	2	1	2	2	2	1	1



x : points de la fonction f
zones grisées : monômes *donnés* pour f
zones encadrées : monômes premiers *obtenus*
-- à coins ronds : monôme premier redondant (consensus des deux autres)
-- à coins carrés : monômes premiers de la base irredondante

Fig. 15.9 Diagramme de Veitch-Karnaugh correspondant.

15.6.6 Bases minimales

S'il existe plusieurs bases irredondantes, entre elles se pose la question des ***bases de coût minimal***: cette phase réintroduit si nécessaire des données technologiques.

Considérons le cas ci-après où une fonction à 8 points mène à 8 monômes premiers à n – 1 variables (tab. 15.8).

Couvrir tous les points peut s'écrire en termes d'expression booléenne :

jklmnpqr = A(A+B+H)(B+C)(C+D)(D+E+F)E(F+G)(G+H)
= A(B+C)(C+D)E(F+G)(G+H) /par absorption[9] : x(x + y) = x
= A(C+BD)E(G+FH) /par commutativité et distributivité

D'où 4 couvertures irredondantes :

ACEG, ACEFH, ABDEG, ABDEFH

On peut ici *présumer* ACEG minimale (fig. 15.10).

[9] La réalisation de contraintes fortes règle en passant des contraintes faibles apparentées ; ici, j rend A obligatoire, et de ce fait la clause (A+B+H) est satisfaite, car A couvre aussi k ; de même, p rend E obligatoire, et n est de ce fait réalisé.

Tableau 15.8 Couverture des points d'une fonction (cas 2).

Termes réalisateurs	Points à réaliser							
	j	k	l	m	n	p	q	r
A obligatoire	x	x						
B		x	x					
C			x	x				
D				x	x			
E obligatoire					x	x		
F					x		x	
G							x	x
H		x						x
	1 critique	3 (A)	2	2	3 (E)	1 critique	2	2

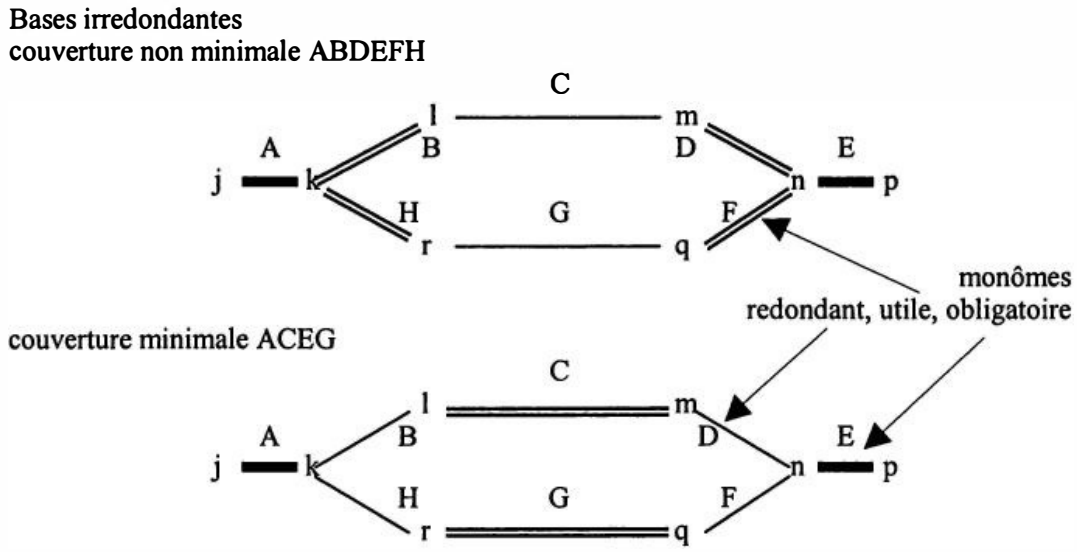


Fig. 15.10 Bases irredondantes : représentation graphique.

15.6.7 Impact technologique

Il n'y a pas de technologie imposant par nature les calculs en et/ou/non, mais ce triplet correspond à un compromis confortable entre faisabilité des calculs et réalisme technique.

Une technologie binaire sera donc *admissible* si elle permet de réaliser « et/ou/non », car par eux toutes les fonctions booléennes seront faisables, et, à des codages binaires près, toute fonction discrète.

Certains théoriciens ont considéré « et/ou/non » comme surabondants, se limitant par exemple au couple « ou/non ».

« Et » apparaît alors comme un opérateur secondaire défini par

$$et(x, y) \equiv non(ou(non(x), non(y))$$

Au début de l'emploi des triodes (années 40) ou des transistors (années 60), la physique a encouragé le minimalisme. Reprenant des travaux tels que ceux de Sheffer, Pierce, Ivanescu, on a tout centré sur le « ni » (resp. le « on »), directement réalisables à partir d'un élément amplificateur, triode ou transistor simple.

Ces *opérateurs de base* laissaient la possibilité de raisonner en termes de « et/ou/non », du fait des relations données tableau 15.9, qui montrent qu'une synthèse effective s'obtient en démarquant les schémas « et/ou/non », double couche par double couche (la fonction spécifiée figure 15.5 est ainsi réalisée selon la figure 15.11).

Tableau 15.9 Opérateurs minimaux « ni/on ».

$ni(x, y, z) \equiv (x + y + z)' = x'y'z'$	**$on(x, y, z) \equiv (xyz)' = x' + y' + z'$**
$ni(x, x, x) = x' \equiv non(x)$	**$on(x, x, x) = x' = non(x)$**
$ni(x, x, x') = 0$, $ni(0,0,0) = 1$	**$on(x, x, x') = 1$, $on(1, 1, 1) = 0$**
$ni(non(x), non(y), non(z)) = xyz = et(x,y,z)$	**$on(non(x), non(y), non(z)) = x + y + z = ou(x, y, z)$**
$non(ni(x, y, z)) = x + y + z = ou(x, y, z)$	**$non(on(x, y, z)) = x \cdot y \cdot z = et(x, y, z)$**
$ni(ni(p, q, r), ni(s, t, u), ni(v, w, x)) =$	**$on(on(p, q, r), on(s, t, u), on(v, w, x)) =$**
$non(ni(p, q, r)) \cdot non(ni(s, t, u)) \cdot non(ni(v, w, x)) = ou(p, q, r) \cdot ou(s, t, u) \cdot ou(v, w, x) =$	$non(on(p, q, r)) + non(on(s, t, u) + non(on(v, w, x)) = et(p, q, r) + et(s, t, u) + et(v, w, x) =$
$(p + q + r) \cdot (s + t + u) \cdot (v + w + x))$	**$p \cdot q \cdot r + s \cdot t \cdot u + v \cdot w \cdot x$**
» une *double couche de NI* réalise des *produits de sommes*	» une *double couche de ON* réalise des *sommes de produits*

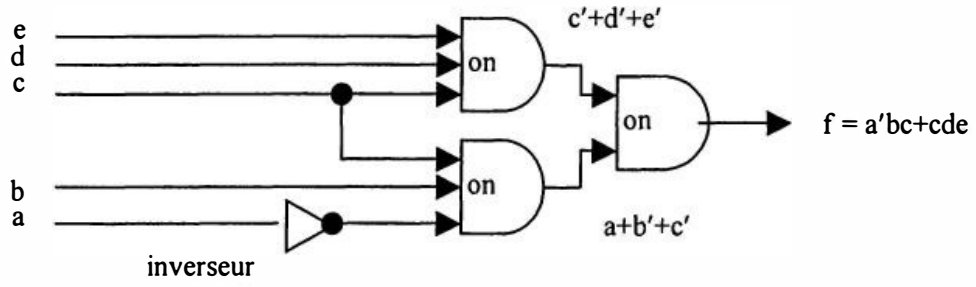


Fig. 15.11 Réalisation d'une fonction booléenne par double couche ON = NON(ET()), dérivée d'une somme de produits (cf. fig. 15.5).

Au-delà de ces questions d'admissibilité, l'efficience est obtenue depuis les années 70 par l'utilisation d'unités fonctionnelles spécialisées à un plus haut niveau, basées cette fois sur :

- le niveau d'intégration disponible,

- la demande, en termes de fréquence et de débit marchand,

unités qui tendent à simplifier les fabrications en les stéréotypant de plus en plus.

Cependant, les fabrications en petites séries comme les ruptures technologiques supposent toujours de savoir revenir aux fondamentaux.

NOTE. Ceux-ci permettent même d'expliquer certains aspects des neuro-sciences. En 1941, Pitts et McCulloch ont suggéré de formaliser les ***neurones*** par des opérateurs semi-analogiques dits ***outils à seuil***. Ces outils ont des entrées et une sortie binaire, sortie valant 1 ssi la somme des entrées dûment pondérées atteint ou dépasse un certain seuil.

$OS(e_1, e_2, \dots e_n)_{p1, p2, \dots pn\,;\,seuil} \equiv (\text{si } p_1e_1+p_2e_2+ \dots + p_ne_n \geq seuil \text{ alors } 1 \text{ sinon } 0).$

On montre facilement que :

$OS(e_1, e_2, \dots e_n)_{1, 1, 1, \dots 1\,:\,1} \equiv OU(e_1, e_2, \dots e_n)$

$OS(e_1, e_2, \dots e_n)_{1, 1, 1, \dots 1\,:\,n} \equiv ET(e_1, e_2, \dots e_n)$

$OS(e_1, e_2, \dots e_n)_{-1, -1, -1, \dots -1\,:\,0} \equiv NI(e_1, e_2, \dots e_n) \dots$

Sans préjuger d'autres combinaisons intéressantes, notamment en correction d'erreurs, ceci suffit à suggérer pourquoi les neurones sont capables de traitement d'information.

15.6.8 Cas des fonctions incomplètes

Ces techniques resteraient souvent d'un intérêt limité si on n'avait à traiter que des fonctions complètes. En fait, de nombreuses fonctions booléennes à réaliser ne sont pas complètement spécifiées car certaines combinaisons de leurs variables sont impossibles.

EXEMPLES

- Pour satisfaire notamment les comptables, pour les affichages etc. la question se pose d'avoir des systèmes ou des composants travaillant en décimal ; de fait ils utilisent souvent des codes binaires non saturés tels que les différents codes « Décimal Codé Binaire » :

  - Le principe des codes *pondérés* est le suivant : un quadruplet booléen abcd représente un chiffre décimal x selon le code pqrs ssi ap + bq + cr + ds = x, un code pqrs étant admissible s'il permet d'atteindre tout chiffre décimal x. Ces codes pondérés sont nombreux, du 8421 au 3321, sans compter la possibilité de poids négatifs. Dans le cas du 8421, les dix codes de 0000 à 1001 ont une interprétation de 0 à 9, tandis que les six codes 1010 à 1111 n'en ont pas.

  - On peut envisager des codes encore moins compacts, comme le code pondéré 22221, ou le code à 5 bits associant à chaque chiffre décimal une combinaison formée de deux bits à 1 parmi 5, ou le code Johnson {00000, 00001, 00011, 00111, 01111, 11111, 11110, 11100, 11000, 10000}. Pour ces codes à 5 bits, seuls 10 codes ont besoin d'une interprétation décimale, et donc 22 peuvent être sans interprétation.

  Au sens de ces codes, la réalisation binaire d'une fonction parfaitement définie au niveau décimal ne sera donc souvent que partiellement définie au niveau binaire, du fait des combinaisons binaires sans interprétation décimale.

- En électronique digitale, en automatisme, on tend à créer autant de variables que nécessaire, chacune ayant un rôle précis ; cela accroît en principe la lisibilité du système, mais il faut s'attendre à des interdépendances rendant impossibles de nombreuses combinaisons, ce qui permettra de simplifier les fonctions de décision correspondantes.

Ces fonctions sont le plus souvent traitées comme applications de $\mathbb{B}^n$ dans $\mathbb{T} = \{0, \phi, 1\}$, où ϕ est affecté aux cas indifférents/indéterminés.

$\mathbb{T}$ est doté d'un système compatible avec celui de $\mathbb{B}$, défini dans le tableau 15.10. La principale nouveauté réside dans l'apparition d'une relation $\sim$ de compatibilité entre valeurs, réflexive, symétrique mais non transitive, telle que $0 \sim \phi$, $\phi \sim 1$, mais $0 \not\sim 1$.

Tableau 15.10 Opérateurs de **T**.

x	x'
0	1
ϕ	ϕ
1	0

+	0	ϕ	1
0	0	ϕ	1
ϕ	ϕ	ϕ	1
1	1	1	1

·	0	ϕ	1
0	0	0	0
ϕ	0	ϕ	ϕ
1	0	ϕ	1

$\leq$	0	ϕ	1
0	1	1	1
ϕ	0	1	1
1	0	0	1

$\sim$	0	ϕ	1
0	1	1	0
ϕ	1	1	1
1	0	1	1

DÉFINITION 15.9 – L'ensemble G des applications de $\mathbb{B}^n$ dans $\mathbb{T}$ est muni d'un ***ordre de réalisabilité*** $\sqsubseteq$ défini par :

$$f \sqsubseteq g \equiv \forall x \in \mathbb{B}^n\, (f(x) = 0 \rightarrow g(x) = 0) \wedge (f(x) = 1 \rightarrow g(x) = 1)$$

Si $f \sqsubseteq g$, f et g sont compatibles ; comme g reprend les cas déterminés de f, auxquels elle peut en ajouter un ou plusieurs, on dira que *f spécifie g* ou *g réalise f.*

PROPOSITIONS

15.8 – $\sqsubseteq$ est un *ordre large*, qui dote cet ensemble G d'une structure d'inf-demi-treillis ayant pour élément minimum la fonction totalement indéterminée ϕ, et pour éléments maximaux les fonctions booléennes totalement définies.

15.9 – *Une fonction f indéterminée pour k n-uples est réalisable par 2^k fonctions g déterminées compatibles : pour réaliser f, il suffira de réaliser l'une de ces fonctions g*

... ce qui ouvre encore le champ des optimisations.

15.10 – Parmi toutes les fonctions g déterminées compatibles avec une fonction partielle f, il existe 2 fonctions $\underline{f}$ et $\overline{f}$ extrêmes au sens de $\leq$[10].

f est ainsi *encadrée* par ces 2 applications $\underline{f}$ et $\overline{f}$ de $\mathbb{B}^n$ dans $\mathbb{B}$, définies par :

$$\underline{f}(x, y, z) \equiv (\text{si } f(x, y, z) = 1 \text{ alors } 1 \text{ sinon } 0)$$

$$\overline{f}(x, y, z) \equiv (\text{si } f(x, y, z) = 0 \text{ alors } 0 \text{ sinon } 1)$$

La double relation $\underline{f} \leq f \leq \overline{f}$ peut aussi s'interpréter $f = \underline{f} + \phi \cdot \overline{f}$.

Les méthodes précédentes s'appliquent au mieux aux fonctions incomplètes, à condition que la couverture finale détermine les bases irredondantes formées de monômes premiers de $\overline{f}$ qui couvrent les points de $\underline{f}$.

[10] Si nécessaire, ces fonctions seront notées respectivement inf(f) et sup(f).

EXEMPLE. On considère le codage binaire 8421 d'un chiffre décimal, et on veut réaliser une fonction booléenne f valant 1 quand le chiffre est divisible par 3 (0 inclus).

On a f = {0, 3, 6, 9} + ϕ{10 et plus} en termes décimaux et, en passant aux codes 8421,

f = {0000, 0011, 0110, 1001} + ϕ {1010, 1011, 1100, 1101, 1110,1111}
$\underline{f}$ = {0000, 0011, 0110, 1001}, *insimplifiable en soi,* menant à un schéma de *coût 16*
$\overline{f}$ = {0000, 0011, 0110, 1001, 1010, 1011, 1100, 1101, 1110,1111}
= {0000, x011, x110, 1xx1, 1x1x, 11xx}

Sont totalement *inutiles* dans $\overline{f}$ les monômes ne couvrant que la partie ϕ, tels que 101x (cas 10,11) et 11xx (cas 12, 13, 14, 15).

Tableau 15.11 Couverture d'une fonction f incomplète.

base de $\overline{f}$		points de $\underline{f}$			
		0000 /0	0011 /3	0110 /6	1001 /9
0000	obligatoire	x			
x011	obligatoire		x		
x110	obligatoire			x	
1xx1	obligatoire				x
1x1x	inutile				
11xx	inutile				

Sont ici *obligatoires* 0000 pour 0, x011 pour 3 (auquel on joint 11 qui ne se produira jamais), x110 pour 6 (auquel on joint 14 qui ne se produira jamais), 1xx1 pour 9 (annexant les impossibles 11, 13, 15). Tous les points de $\underline{f}$ sont couverts : aucun autre monôme de $\overline{f}$ n'est utile.

D'où une *base irredondante de coût 12 :* {0000, x011, x110, 1xx1}.

Parmi les *monômes d'un base première de* $\overline{f}$ on pourra donc distinguer :

- les *obligatoires*, seuls à couvrir au moins un point de $\underline{f}$;
- les *inutiles*, ne couvrant que des points ϕ ou des points couverts par les obligatoires ;
- les *utiles*, couvrant (non exclusivement) au moins un point de $\underline{f}$ non couvert par les obligatoires,
- les *redondants* enfin.

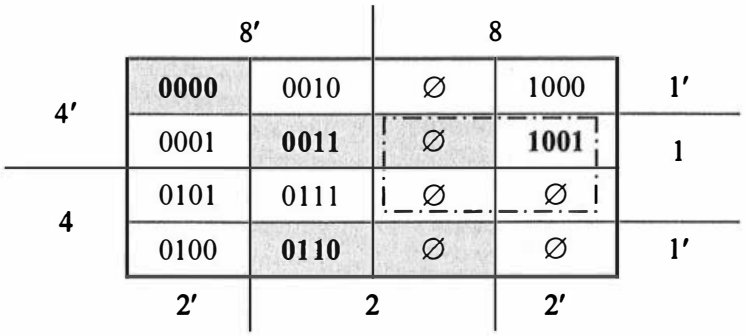


Fig. 15.12 Diagramme de Veitch-Karnaugh. Optimisation par *annexion d'indéfinis*.

15.7 DES SOMMES DE MONÔMES AUX PRODUITS DE CLAUSES

Etant donné la dualité de l'algèbre de Boole, reflétée par exemple par les règles d'Occam/de Morgan, tout pourrait être redit en permutant + et ·, 0 et 1... Mais si la dualité enrichit nos procédés, elle nous amène surtout à aborder de nouvelles questions.

En effet, tandis que *l'électronique digitale* privilégiait l'expression des fonctions booléennes en *somme de produits*, les calculs booléens liés à la *démonstration automatique* et à l'*intelligence artificielle* préféraient transcrire les ***bases de connaissances*** sous la forme duale de ***produits de clauses***, *amorçant ainsi une autre pratique.*

15.7.1 Duale d'une fonction

DÉFINITION 15.10 – *La **duale d'une fonction** $f(x, y, \ldots, z)$ est la fonction f^* telle que*

$$f^*(x, y, \ldots, z) \equiv f'(x', y', \ldots, z')$$

REMARQUE. De $(a + b + c)' = a'b'c'$ on tire $(a + b + c)^* = abc$. De même de $(abc)' = a' + b' + c'$ on tire $(abc)^* = a + b + c$.

Par suite, f^* s'obtient à partir de f en permutant + et . sans autre effort : f^* est en général plus facile à calculer que f'.

PROPOSITION 15.11 – La dualité est involutive : $f^{**} = f$.

PREUVE. $f^{**}(x, y \ldots z) = (f^*(x, y \ldots z))^* = (f'(x', y' \ldots z'))^* = (f''(x'', y'' \ldots z'')) = f(x, y \ldots z)$, car $x'' \equiv x$ et $f'' \equiv f$.

DÉFINITION 15.11 – Si $f = f^*$, la fonction f est dite ***fonction autoduale***, sinon f et f^* sont dites ***fonctions duales par paires.***

EXEMPLES

- et/ou, ni/on sont *duales par paires.*
- $maj(x, y, z) = xy + xz + yz$ est *autoduale.*

PROPOSITION 15.12 – Tout dispositif booléen qui réalise f dans une technologie donnée sous une convention physique précise pour 0 et 1, réalise f^* sous la convention inverse.

PROPOSITION 15.13 – Le rôle d'un dispositif réalisant une fonction f autoduale dans une technologie donnée sous une convention physique précise pour 0/1, est inchangé sous la convention inverse.

15.7.2 Notion de clause

DÉFINITION 15.12 – Nous appellerons

- ***clause***[11] le dual d'un monôme, i.e. une *somme de variables*, chacune sous forme directe ou complémentée ;
- ***clause canonique,*** une clause où chaque variable est présente, sous forme directe ou complémentée.

[11] Monal au sens de Kuntzmann ; nous abandonnons cette terminologie pour éviter les confusions verbales entre monaux et monômes.

15.7.3 Seconde forme canonique

PROPOSITION 15.14 – *Toute fonction booléenne f peut s'écrire comme produit de clauses canoniques,* ce qui introduit une ***seconde forme canonique*** (FC2).

PREUVE 1. Toute fonction booléenne peut être écrite comme somme de monômes canoniques (FC1, prop. 15.3).

Soit une fonction f quelconque : son complément f' peut être écrit comme somme de monômes canoniques : $f' = \Sigma\mu$; par conséquent, f peut s'écrire $f = f'' = \Pi\mu'$, où les μ', compléments de monômes canoniques, sont des clauses canoniques. ■

PREUVE 2. Toute fonction booléenne peut être écrite comme somme de monômes canoniques (FC1, prop. 15.3).

Soit une fonction f quelconque : sa duale f* peut être écrite comme somme de monômes canoniques : $f^* = \Sigma\mu$; par conséquent, f peut s'écrire $f = f^{**} = \Pi\mu^*$, où les μ^*, duaux de monômes canoniques, sont des clauses canoniques. ■

EXEMPLE. Définir la FC2 d'une fonction booléenne détectant les quadruplets valant 5 et plus dans un code 8421.

Soit f la fonction cherchée. Son complément est défini par f' ={0000, 0001, 0010, 0011, 0100}, (codes de 0 à 4)

soit $f'(a, b, c, d) = a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + a'bc'd'$

On a toujours $(pqrs)' = p' + q' + r' + s'$

d'où $f(a, b, c, d) = (a+b+c+d)\cdot(a+b+c+d')\cdot(a+b+c'+d)\cdot(a+b+c'+d')\cdot(a+b'+c+d)$.

C'est la FC2 cherchée.

15.7.4 Bases clausales

La distributivité entraîne que $(a + x)\cdot(a + x') = a + x\cdot x' = a$, et l'absorption que $(s + x)\cdot s = s + x\cdot s = s$.

On peut donc s'attendre à des produits de sommes dont les sommes ou clauses ne contiennent pas toutes les variables.

EXEMPLE. Considérons la fonction booléenne détectant les quadruplets valant 5 et plus dans un code 8421.

Sous sa seconde forme canonique, elle s'écrit

$$f(a, b, c, d) = \underset{①}{(a+b+c+d)}\cdot\underset{②}{(a+b+c+d')}\cdot\underset{③}{(a+b+c'+d)}\cdot\underset{④}{(a+b+c'+d')}\cdot\underset{⑤}{(a+b'+c+d)} \qquad / \text{cout } 20$$

$$f(a, b, c, d) = \underset{①\cdot②}{(a + b + c)\cdot(a + b + c')}\cdot\underset{③\cdot④}{(a + c + d)}\cdot\underset{①\cdot⑤}{} = \underset{①②③④}{(a + b)}\cdot\underset{①⑤}{(a + c + d)} \qquad / \text{cout } 5$$

Par dualité du consensus, nous avons :

PROPOSITION 15.15 – (Quine, Robinson) *Règle de résolution :*

$$(a + p)\cdot(a' + q) = (a + p)\cdot(a' + q)\cdot(p + q)$$

NOTE. La clause (p + q), dite ***clause résolvante*** *des clauses (a + p) et (a'+q),* est exploitable en démarquant point par point la méthode des consensus.

Bases clausales et listes de propositions

Outre la dualisation des problèmes précédents, *les **bases clausales** apparaîtront aussi comme codage algébrique d'un ensemble d'énoncés propositionnels.*

Considérons en effet un système, possédant un état et des lois, comme décrit par une conjonction de faits (généralement variables propositionnelles isolées, vraies sous l'une ou l'autre forme) et de règles ou implications.

Les règles de passage logique propositionnelle / calcul booléen sont fondées sur la mise en équivalence

- de l'implication $a \wedge b \wedge c \wedge d \rightarrow x$ *(si a et b et c et d alors x)*,
- avec son interprétation philonienne $\neg (a \wedge b \wedge c \wedge d \wedge \neg x)$ soit $(a \cdot b \cdot c \cdot d \cdot x')'$,
- et donc avec la clause $a' + b' + c' + d' + x$.

Une conjonction d'implications se ramène ainsi à un produit de clauses. Qu'il s'agisse d'une base de règles et/ou d'une base de faits :

PROPOSITION 15.16 – *Une base de connaissances d'ordre 0 peut être représentée par une base clausale.*

Or, à la formation des bases de connaissances, la priorité porte sur l'*exactitude* des connaissances formalisées, et non sur l'habileté de cette formalisation. Les sections antérieures suggèrent que les *bases de connaissances seront souvent optimisables* ; de plus, les méthodes d'optimisation pourront accessoirement *détecter des incohérences.*

Bases clausales premières

PROPOSITION 15.17 – Est ***clause première*** toute clause d'une base, dont aucune sous-clause n'est aussi clause de la base.

En ce sens, une clause première est irréductible.

En termes de bases de connaissances, il s'agira de *faits isolés, ou d'implications qu'on ne saurait simplifier davantage* sans changer la signification globale de la base de connaissances considérée.

Bases clausales irredondantes, minimales

Considérons une base clausale première B.

DÉFINITION 15.13 – Est ***clause première obligatoire*** toute clause première qu'on ne saurait *retirer* du produit ou de la base sans changer la valeur globale de la base considérée.

Si le produit P_0 des clauses obligatoires équivaut à la base, ce produit est *la* base première irredondante (et minimale).

Sinon, les clauses restantes se partagent en ***clauses utiles*** et ***clauses inutiles ou redondantes***...

Sont *inutiles* les clauses premières non obligatoires C telles que $C \cdot P_0 = P_0$, et sont *utiles* les clauses premières C telles que $B \leq C \cdot P_0 < P_0$. Retenons une de ces clauses utiles, C_1, et formons $P_1 = C_1 \cdot P_0$. Si $P_1 = B$, on a terminé ; sinon, deviennent *redondantes* les clauses restantes C telles que $C \cdot P_1 = P_1 \ldots$, et restent utiles les clauses C_1 telles que $B \leq C \cdot P_1 < P_1 \ldots$

Chaque ***base clausale irredondante*** comprendra, outre les *clauses premières obligatoires,* une sélection de *clauses utiles,* rendant les autres clauses *redondantes* pour cette sélection.

En cas de multiplicité des bases irredondantes, des considérations de coût indiqueront celles de ces bases qui sont minimales.

Exemple

La base de règles ci-après est-elle optimale ?

(i) $a \wedge b \wedge c \rightarrow f$
(ii) $b \wedge c \wedge \neg d \rightarrow a$
(iii) $a \wedge f \rightarrow d$

Sous forme clausale, cette base devient :

$$\begin{aligned} B &= (a' + b' + c' + f)\cdot(b' + c' + d'' + a)\cdot(a' + f' + d) \\ &= (a' + b' + c' + f)\cdot(b' + c' + d + a)\cdot(a' + f' + d) \\ &= (a' + b' + c' + f)\cdot(a + b' + c' + d)\cdot(a' + d + f') \end{aligned}$$

Le passage sous forme clausale a un premier effet de normalisation, une clause à n variables représentant $2^n - 2$ implications apparemment distinctes : à $x \wedge y \rightarrow z$ correspond une clause $(x' + y' + z)$ codant aussi bien $x \wedge \neg z \rightarrow \neg y$, $y \wedge \neg z \rightarrow \neg x$; $x \rightarrow \neg y \vee z$, $y \rightarrow \neg x \vee z$, $\neg z \rightarrow \neg x \vee \neg y$ qu'un trilemme $\neg x \vee \neg y \vee z$ ou un interdit $\neg(x \wedge y \wedge \neg z)$.

Avec un codage additif, la forme clausale permet la production de résolvantes, en bijection avec la production de consensus à partir d'un codage multiplicatif (tab. 15.12) :

Tableau 15.12 Passage d'une base clausale à une base clausale première par le calcul des résolvantes (isomorphe au calcul de la base première d'une somme de produits par la méthode des consensus).

000x1	000x1	000x1	000x1	000x1	000x1	000x1	000x1	000x1
1001x	1001x	1001x	1001x	1001x	1001x	1001x	~~1001x~~	x001x
0xx10	0xx10	0xx10	0xx10	0xx10	0xx10	0xx10	0xx10	0xx10
x0011	x0011	x0011	x0011	x0011	x0011	x0011	~~x0011~~	∅
	0001x	0001x	0001x	0001x	0001x	0001x	~~0001x~~	
		∅	∅	x0010	x0010	x0010	~~x0010~~	
abcdf					∅	x001x		

Les règles i/ et iii/ restant inchangées, la règle ii/ peut être allégée, et on a maintenant :

(i) $a \wedge b \wedge c \rightarrow f$
(ii') $b \wedge c \rightarrow d$ (au lieu de (ii) $b \wedge c \wedge \neg d \rightarrow a$)
(iii) $a \wedge f \rightarrow d$

Si on considère la base en termes d'*interdits élémentaires*, ii' unit 4 interdits élémentaires : les 2 de (ii), 1 de (i) ($abcd'f' = 0$) et 1 de (iii) ($abcd'f = 0$).

15.7.5 Bases premières complètes par double dualisation

PROPOSITION 15.18 – Le développement en somme de produits du produit de *deux* clauses ne comporte que des monômes premiers, et éventuellement certains de leurs multiples (absorbables).

EXEMPLE. $(a + b)\cdot(a + c) = a + ab + ac + bc = a + bc$.

PROPOSITION 15.19 – Le développement en somme de produits du produit de *plusieurs* clauses ne comporte que des monômes premiers, et éventuellement certains de leurs multiples (absorbables).

Claude BENZAKEN a donc proposé en 1966 *la construction de bases premières complètes par double dualisation :*

(a) on a $f^{**} = f$;

(b) si on a f sous forme d'une somme de produits, f se transcrit immédiatement en f^* comme produit de sommes ou clauses ; on effectue ce produit de clauses pour obtenir une somme de monômes, qui, après absorption, sera une base première complète B^* (d'où une base de f' par simple transcription) ;

(c) en dualisant B^*, on obtient f d'abord comme produit de sommes, puis en l'effectuant comme somme de monômes, qui après absorption des multiples sera B, la base première complète de f cherchée.

La beauté de cette méthode est que pour réaliser f, on obtient *en passant* ce qu'il faut pour décider si la réalisation de f par la réalisation de f' puis sa complémentation, ne serait pas moins coûteuse que la réalisation directe de f, avec en prime la disponibilité de f'.

EXEMPLE. Soit une fonction $f = \{00x0, 00x1, 011x, 111x\}$ comprenant 4 arêtes disjointes soit 8 points. Quelle est sa base première ?

Nous avons successivement, en appelant a, b, c, d les variables :

f	$=$	$a'b'd' + a'b'd + a'bc + abc$	*/forme SP initiale de f*
f^*	$=$	$(a' + b' + d')\cdot(a' + b' + d)\cdot(a' + b + c)\cdot(a + b + c)$	/forme PS calquée pour f^*
	$=$	$(a' + b' + d\cdot d')\,(b + c + a\cdot a')$	/on effectue
	$=$	$(a' + b')\cdot(b + c)$	/on réduit
	$=$	$a'b + a'c + b'b + b'c$	/on effectue et réduit
	$=$	$a'b + a'c + b'c$	/B^*, base première de f^*
$\Rightarrow$		$ab' + ac' + bc'$	*/B', base première de f'*
f^{**}	$=$	$(a' + b)(a' + c)(b' + c)$	/forme PS pour f duale de B^*
	$=$	$(a' + bc)(c + a'b')$	/on effectue
	$=$	$a'c + a'b' + bc + a'b'bc$	/on réduit
	$=$	$a'c + a'b' + bc$	*/B**, base première de f ! ! !*

(cf. § 15.6.3)

Cas des fonctions très incomplètes[12]

Pour les fonctions ayant une majorité de cas indéterminés[13], il est commode d'organiser les traitements à partir de

- inf(f), liste des 1 de la fonction f,
- inf(f'), liste des 0 de la fonction f.

Si inf(f') est sous forme SP (ex : première forme canonique), la base première de sup(f) s'obtient facilement en effectuant (inf(f'))' : la forme SP de inf(f') se transcrit directement en forme PS de (inf(f'))' ; l'évaluation de celle-ci, après absorption des monômes absorbables, donne la base première complète de sup(f) sans avoir à définir autrement cette fonction (cf. exemple en 15.7.3).

$\underline{f}$ = inf(f) fournit alors les points à couvrir.

EXEMPLE. Soit inf(f) = {0001, 0010, 0100, 1000} et inf(f') = {1110, 1101, 1011, 0111} ; posons inf(f') = abcd' + abc'd + ab'cd + a'bcd ;

$$(\inf(f'))' = (a' + b' + c' + d)\cdot(a' + b' + c + d')\cdot(a' + b + c' + d')\cdot(a + b' + c' + d')$$
$$= (a' + b' + cd + c'd')\cdot(a'b' + ab + c' + d'')$$
$$= \underset{①}{a'b'} + \underset{②}{a'c'} + \underset{③}{a'd'} + \underset{④}{b'c'} + \underset{⑤}{b'd'} + \underset{⑥}{c'd'} ;$$

inf(f) est couverte par (①+②+④) pour 0001, (①+③+⑤) pour 0010, (②+③+⑥) pour 0100 et (④+⑤+⑥) pour 1000,

soit comme couvertures (①+②+④)·(①+③+⑤)·(②+③+⑥)·(④+⑤+⑥) = ①⑥ + ②⑤ + des combinaisons à 3 termes

soit ①⑥= a'b' + c'd' et ②⑤= a'c' + b'd' comme bases minimales.

15.8 APPROCHES DICHOTOMIQUES

KUNTZMAN, AKERS, BRYANT, entre autres, ont plaidé l'intérêt de synthèses ou de spécifications booléennes à partir d'un opérateur duplex U(x, y, z) = x'y + xz, exploitant la relation :

$$f = U(x, f|_{x=0}, f|_{x=1}) \text{ (BOOLE, Shannon)}$$

La première idée est de s'en servir pour une synthèse arborescente, chaque niveau de décision éliminant une variable, ramenant la synthèse d'une fonction à n variables à la synthèse de 2 fonctions à n – 1 variables.

Pour f = {1, 2, 3, 5, 7, 11, 13}, les variables a, b, c, d étant attachées aux poids 8, 4, 2, 1, on a :

$$\begin{aligned} f &= a'b'c'd + a'b'cd' + a'b'cd + a'bc'd + a'bcd + ab'cd + abc'd \\ &= U(a, b'c'd + b'cd' + b'cd + bc'd + bcd\,,\, b'cd + bc'd) \\ &= U(a, U(b, c'd + cd' + cd, c'd + cd), U(b, cd, c'd)) \\ &= U(a, U(b, U(c, d, d' + d), U(c, d, d)), U(b, U(c, 0, d), U(c, d, 0))) \ldots \end{aligned}$$

---

[12] Voir aussi KUNTZMANN, *Algèbre de Boole*, pp. 28 et seq.

[13] Techniquement fréquentes : si on doit combiner 2 chiffres décimaux codés en DCB, il y a 100 cas de figures définis pour 256 cas théoriques à partir des codes à 4 bits, soit 60% de cas indéterminés...

Dans l'arbre brut obtenu, à chaque monôme canonique correspond un chemin et un seul, d'une constante 1 à la racine (sortie). Pour n variables, il faudrait ainsi $1 + 2 + 4 + \ldots + 2^{n-1} = 2^n - 1$ duplex.

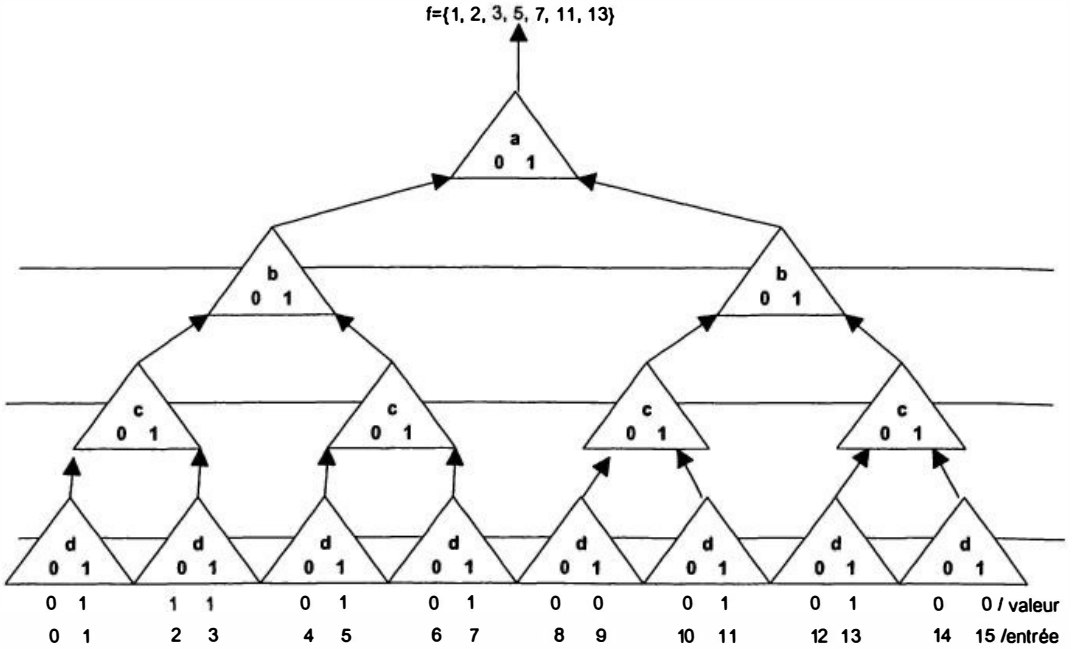


Fig. 15.13 Arbre brut d'une fonction booléenne f.

On passe de cet arbre brut à un *arbre réduit* en remplaçant tout outil U menant à deux sous-arbres identiques par un seul de ces sous-arbres, car U(x, A, A) = A.

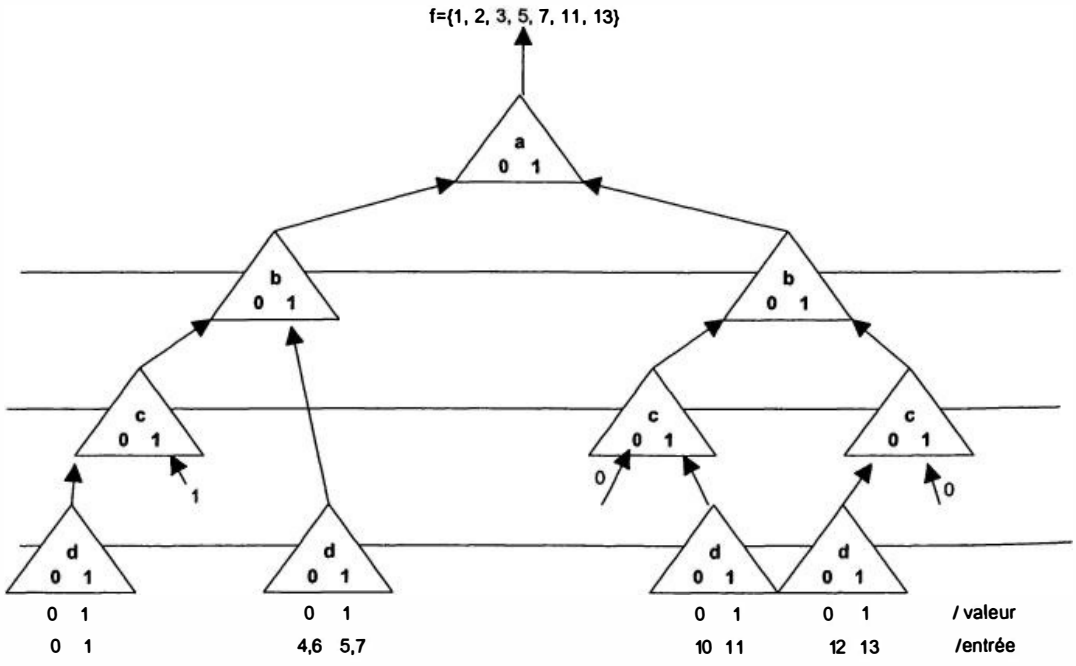


Fig. 15.14 Arbre réduit de la fonction booléenne f.

On passe de cet arbre réduit à un *graphe réduit* si on remplace chaque paire de sous-graphes identiques par deux connexions à la sortie d'un seul de ces sous-graphes.

Dans un tel graphe, à chaque monôme canonique correspond un chemin d'une constante 1 à la sortie, et à chaque chemin correspond un monôme, disjoint des monômes correspondant aux autres chemins.

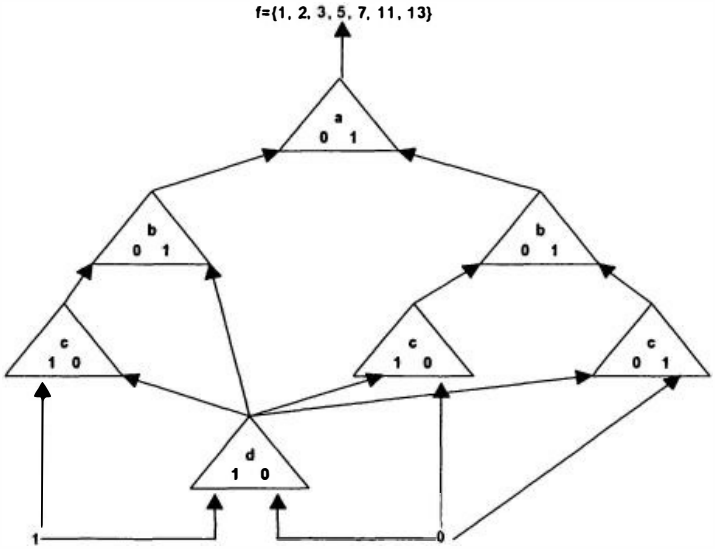


Fig. 15.15 Graphe réduit de la fonction booléenne f.
Attention : des entrées (0 1) ont été permutées (1 0) pour éviter les croisements.

Un graphe réduit sera dit *graphe canonique* si c'est un graphe réduit au maximum, les variables apparaissant dans un ordre imposé.

Le graphe est *multifonctionnel* s'il porte sur l'élaboration simultanée de plusieurs fonctions booléennes, ce qui modélise bien

- les circuits qui élaborent simultanément plusieurs fonctions dépendant des mêmes variables (ex : additionneurs...),
- la prise de décisions multiples interdépendantes.

Ce formalisme des *diagrammes de décision binaire* [AKERS, BRYANT] est souvent apprécié pour plusieurs raisons :

- il fournit des définitions exactes qui possèdent une forme canonique ;
- il est assez léger pour les fonctions symétriques ;
- il permet de traiter de front plusieurs fonctions ;
- il peut participer à la description de systèmes binaires très complexes, par l'introduction de *niveaux de description* et d'actions *plier/déplier* ;
- il spécifie aussi bien des circuits binaires (dans une technologie non précisée) que des systèmes de décision.

Il est très employé en matière de test et de simulation.

En conception, il peut être utilisé surtout comme spécification. Mais pour les couches les plus physiques, certaines fonctions mènent à des graphes réduits de taille trop sensible à l'ordre imposé aux variables pour que ces définitions servent de descriptions.

L'extension à des fonctions à valeur dans $\mathbb{T} = \{0, \varnothing, 1\}$ permet de tenir compte d'interdépendances entre variables, ou d'examiner les régimes transitoires dans les systèmes booléens sous étude.

15.9 ÉQUATIONS BOOLÉENNES

Qu'il s'agisse de *bases de connaissances à l'ordre 0*, des interdépendances entre variables logiques d'un dispositif ou d'un logiciel, ou du codage des données d'une énigme, on peut ramener les relations et contraintes

- soit à une collection de relations $f_i = 0$, qu'on traite en bloc sous la forme $\Sigma f_i = 0$,
- soit à une collection de relations $f_i = 1$, qu'on traite en bloc sous la forme $\Pi f_i = 1$.

Les sections précédentes ont montré comment alléger et condenser ces formes globales sans perte d'information. On considérera le résultat comme invariant du système logique. Quelle est la réponse à telle ou telle question dans ce contexte ?

Si I est l'invariant du système et x la question, x peut être connu en résolvant une équation I(x). Outre une réponse pour x, on pourra avoir une *indétermination* (l'état du système ne suffit pas à définir x) ou une *impossibilité* (l'état du système est contradictoire / invalide, et x ne peut être).

15.8.1 Equation booléenne a·x = b

En examinant les 4 cas possibles, nous pouvons dire que :

PROPOSITION 15.20 – L'équation $a \cdot x = b$ admet une solution si $a'b = 0$, soit $b \leq a$.

Cette solution est $x = b + \phi a'$, ou, sous forme paramétrique, $x = b + u \cdot a'$, où u est une variable booléenne libre.

Tableau 15.13 Cas pour l'équation a·x = b.

ab	a·x = b	cas	x
00	0x = 0	indéterminé	ϕ
01	0x = 1	impossible	///
10	x = 0	déterminé	0
11	x = 1	déterminé	1

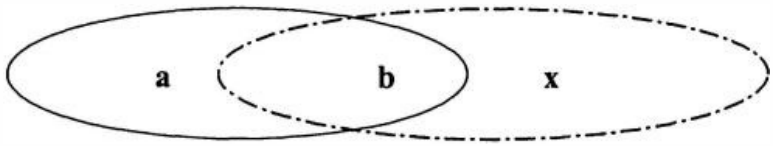


Fig. 15.16 Diagramme pour a·x = b.

EXEMPLE. Si Jean est au violon, et Léon à l'accordéon, tout le monde danse la polka, et seulement dans ce cas. Que fait Léon ?

- Si tout le mode danse la polka, Jean est au violon et Léon à l'accordéon.
- Si on ne danse pas la polka,
  - ou Jean est au violon, et Léon n'est pas à l'accordéon,
  - ou Jean n'est pas au violon, et ce que fait Léon est indéterminé.

15.9.2 Equation booléenne a·x + b·x′ = c

On peut se ramener à l'équation précédente (§ 15.9.4) ou examiner directement les 8 cas de figure, et en déduire :

PROPOSITION 15.21 – L'équation a·x + b·x′ = c

- admet une solution si a′b′c + abc′ = 0 soit ab ≤ c ≤ a + b,
- cette solution est alors x = a′bc′ + ab′c + t(ab + a′b′), où t est une variable booléenne libre ; ou, si on préfère, a′bc′ + ab′c ≤ x ≤ (a + b′ + c′)(a′ + b + c).

Tableau 15.14 Etude de l'équation a·x + b·x′ = c.

abc	ax + bx′ = c	cas	x
000	0x + 0x′ = 0	indéterminé	ϕ
001	0x + 0x′ = 1	impossible	///
010	0x + 1x′ = 0	x′ = 0	1
011	0x + 1x′ = 1	x′ = 1	0
100	1x + 0x′ = 0	x = 0	0
101	1x + 0x′ = 1	x = 1	1
110	1x + 1x′ = 0	impossible	///
111	1x + 1x′ = 1	indéterminé	ϕ

récapitulatif pour x

	b′	b	
c′	ϕ	1	a′
c	ϕ	0	
	1	ϕ	a
c′	0	ϕ	

grisé : cas impossibles

EXEMPLE. Quand il pleut les soirs d'hiver, William écoute Jean raconter sa vie, ou bien il joue aux échecs avec Delphine. S'il pleut un soir d'hiver, que fait William si Jean est aphone ? Si Delphine est absente ?

Quand il pleut les soirs d'hiver, Jean et/ou Delphine sont là (condition de validité).

- Si Jean est aphone, Jean ne peut raconter sa vie et William l'écouter, donc William joue aux échecs avec Delphine.
- Si Delphine est absente, William ne peut jouer aux échecs avec elle et il écoute Jean raconter sa vie.
- Il est impossible pour William d'avoir un soir d'hiver Delphine absente et Jean aphone.

Pour les réponses finales, il sera souvent plus parlant de garder la solution sous forme d'inégalités.

Mais si l'équation résolue relève d'un système plus vaste, seule la forme paramétrique dûment reportée permettra de définir exactement les degrés de liberté résiduels pour l'ensemble des variables recherchées.

15.10 ANNEAU DE BOOLE

15.10.1 Présentation

George BOOLE était très soucieux de garder dans ses calculs logiques les propriétés de l'algèbre traditionnelle, notamment en matière de résolution d'équations, approche qu'il privilégiait. De ce fait, ses travaux s'adressent souvent à un dioïde cousin du précédent, menant à une structure d'anneau.

DÉFINITION 15.14 – On appelle ***anneau de Boole*** une structure <𝔹, 0, 1, ⊕, .> formée de l'ensemble 𝔹 = {0, 1} muni des opérations

- ***ou exclusif***, notée ⊕, qui s'interprète arithmétiquement comme une *somme modulo 2,*
- ***et***, notée ·

Tableau 15.15 Opérations ⊕ et ·.

⊕	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

PROPOSITION 15.22 – Soient a, b, c… des variables à valeur dans 𝔹.

Au pire par énumération, on établit les propriétés du tableau 15.16, qui permettent de vérifier que

- <𝔹, 0, 1, ⊕ > est un groupe,
- <𝔹, 0, 1, ⊕, ·> est un anneau,
- et comme < 𝔹 –{0}, ·> = <{1}, ·>se réduit au cas de figure 1·1 = 1, 1 est élément neutre et son propre inverse. <𝔹, 0, 1, ⊕, ·> est un corps !

Comme <𝔹, 0, 1, ⊕, ·> est un corps fini, c'est un corps de Galois, c'est d'ailleurs le plus petit. On le note CG(2).

Tableau 15.16 Propriétés de <𝔹, 0, 1, ⊕, ·>.

propriété	⊕	
associativité	(a⊕b)⊕c = a⊕(b⊕c)	(a · b) · c = a · (b · c)
élément neutre	a ⊕ 0 = a = 0⊕a	a · 1 = a = 1 · a
élément absorbant		a · 0 = 0 = 0 · a
action réflexive	tout élément est involutif a⊕a = 0 *tout élément est son propre inverse*	idempotence a · a = a
simplifiabilité	a⊕c = b⊕c → a = b c⊕a = c⊕b → a = b	
commutativité	a ⊕ b = b⊕a	a · b = b · a
	<𝔹, ⊕, 0, 1 > groupe commutatif	<𝔹, ·, 1, 0 > monoïde idempotent et commutatif
distributivité		a · (b⊕c) = a · b⊕a · c = (b⊕c) · a

PROPOSITION 15.23 – Le passage d'une structure <𝔹, 0, 1, ⊕, ·> à l'autre <𝔹, 0, 1, ′, +, ·> est assuré par les relations

- x′ = 1⊕ x
- x + y = x⊕y⊕x·y, d'où x + y = x⊕y ⇔ x·y = 0
- x ⊕ y = (x + y)·(x′ + y′) = x·y′ + x′·y

15.10.2 Troisième forme canonique

PROPOSITION 15.24 – Toute fonction booléenne f peut s'écrire

- comme *somme par ⊕ de monômes contenant exclusivement des variables sous forme directe,*
- plus éventuellement 1.

Cette forme est unique si les variables dans chaque monôme, et les monômes entre eux, respectent l'ordre lexicographique (FC3).

EXEMPLE. Soit f = abc′ + a′bcd

f = abc′ + a′bcd
= abc′ ⊕ a′bcd /termes disjoints
= ab(1⊕c) ⊕ (1⊕a)bcd /remplacement des variables complémentées
= (ab ⊕ abc)⊕(bcd ⊕ abcd)
= *ab ⊕ abc ⊕ abcd ⊕ bcd*

La représentation des fonctions sous cette forme sera souvent plus simple que les représentations précédentes, car on utilise ici au plus 2^n monômes (contre 3^n monômes ou clauses dans les représentations précédentes). C'est surtout le cas de fonctions de type *clé de parité*, tandis que la représentation des NI devient lourde.

Tableau 15.17 Troisième forme canonique de f= abc′ + a′bcd.

abcd	f	ab	abc	abcd	bcd	Σ	⊕
0000	**0**	0	0	0	0	0	**0**
0001	**0**	0	0	0	0	0	**0**
0010	**0**	0	0	0	0	0	**0**
0011	**0**	0	0	0	0	0	**0**
0100	**0**	0	0	0	0	0	**0**
0101	**0**	0	0	0	0	0	**0**
0110	**0**	0	0	0	0	0	**0**
0111	**1**	0	0	0	1	1	**1**
1000	**0**	0	0	0	0	0	**0**
1001	**0**	0	0	0	0	0	**0**
1010	**0**	0	0	0	0	0	**0**
1011	**0**	0	0	0	0	0	**0**
1100	**1**	1	0	0	0	1	**1**
1101	**1**	1	0	0	0	1	**1**
1110	**0**	1	1	0	0	2	**0**
1111	**0**	1	1	1	1	4	**0**

EXEMPLE. Soit à définir une fonction f valant 1 si x, y, z sont à 1, ou si t, u, v, sont à 1, mais pas les deux. On préfèrera en général f = xyz ⊕ tuv aux formes équivalentes.

SP : f = xyzt′ + xyzu′ + xyzv′ + x′tuv + y′tuv + z′tuv.

PS : f= (x+t)(x+u)(x+v)(y+t)(y+u)(y+v)(z+t)(z+u)(z+v)(x′+y′+z′+t′+u′+v′).

15.10.3 Traitement de l'hypothèse du monde clos

Qu'advient-il des bases de connaissances à l'ordre 0 lorsqu'on veut prendre en compte l'***hypothèse du monde clos*** ?

Rappelons qu'une base de connaissances telle que
$a \wedge b \rightarrow z$; $a \wedge c \rightarrow z$ était assimilée jusque là à
$a \cdot b \leq z$; $a \cdot c \leq z$ codé $(a' + b' + z) \cdot (a' + c' + z) = 1$, soit $a' + b'c' + z = 1$.
Si $a = 1$ et $b + c = 1$ alors on a bien $z = 1$.

Mais si $a = 0$ ou $b + c = 0$, $1 + z = 1$, et *z est indéterminé : au sens des connaissances identifiées, z est faux, mais quelque autre cause inconnue pourrait le rendre vrai.*

Avec l'*hypothèse du monde clos*, z ne peut connaître d'autre cause que celles données et $z \geq a \cdot b + a \cdot c$ *devient* $z = a \cdot b + a \cdot c$. Pour calculer l'invariant de cette base, passons aux $\oplus$ qui permettront un changement de membre pour z ; $z = ab \oplus ac \oplus abc$, donc $z \oplus ab \oplus ac \oplus abc = 0$[14].

Si $a = b = 1$, $z \oplus 1 \oplus c \oplus c = z \oplus 1 = 0$ et $z = 1$, comme cherché. Si $b = c = 0$, $z \oplus 0 \oplus 0 \oplus 0 = z = 0$, *et aucune cause cachée ne peut plus entraîner z = 1.*

L'approche *anneau de Boole* facilite ainsi le *traitement de l'hypothèse du monde clos*, si on considère un paquet de règles comme définissant une *équivalence* entre une proposition et le système de ses causes identifiées.

15.10.4 Apport aux équations booléennes

Elles sont maintenant plus faciles à traiter, car *$\oplus$ autorise les simplifications membre à membre et les changements de membre,* étant à la fois addition et sa propre soustraction.

PROPOSITION 15.25 – $a \oplus x = b$ a pour solution $x = a \oplus b$.

PREUVE. Par addition membre à membre de a. ■

PROPOSITION 15.26 – De même $ax \oplus b = c$ devient $ax = b \oplus c$.

PROPOSITION 15.27 – L'équation $ax + bx' = c$ se ramène à la forme $a_1x = b_1$.

PREUVE. $ax + bx' = ax \oplus bx' = ax \oplus b \oplus bx = (a \oplus b)x \oplus b$ d'où $ax + bx' = c \Rightarrow ax + bx' = (a \oplus b)x \oplus b = c \Rightarrow (a \oplus b)\, x = b \oplus c$, et la seconde forme d'équation se ramène à la première. ■

COROLLAIRE 15.28 – Comme $Ax = B$ suppose toujours $B \leq A$ et donne $x = B + uA'$ soit $x = B \oplus uA' = B \oplus u \oplus uA$, on en déduit

- que $ax + bx' = c$ a pour solution $x = b \oplus c \oplus u(1 \oplus a \oplus b)$
- dès lors que $(a \oplus b)(b \oplus c) = b \oplus c$ soit $(1 \oplus a \oplus b)(b \oplus c) = 0$.

15.11 ÉQUIVALENCES MAJEURES

Logique propositionnelle (chap.1), *théorie des ensembles* (chap. 2) et *calcul booléen* peuvent être mis en équivalence à l'aide de la table ci-dessous. Il y a donc avantage à *considérer l'algèbre de Boole comme catégorie* (§ 12.4.5) plutôt que comme système spécifique.

---

[14] Cette forme facilite l'intégration des différents paquets de règles relatifs aux autres propositions.

Tableau 15.18 Equivalences majeures entre algèbres de Boole.

Ensembles	Propositions	Calcul booléen
Référentiel	Vrai	1
ϕ	Faux	0
A	a	a
$A \subseteq B$	$a \rightarrow b$	$a \leq b$
$\cap$	$\wedge$	$\cdot$
$\cup$	$\vee$	+
Δ	$\vee\vee$	$\oplus$
Ref–A	$\neg a$	a′
A–B	$a \wedge \neg b$	ab′
=	$\leftrightarrow$	=

15.12 EXERCICES

15.12.1

Montrer que $x + y = x \oplus y \oplus x \cdot y$.

15.12.2

On s'intéresse au codage de chiffres décimaux sur 4 bits (a, b, c, d) de telle façon que la valeur du chiffre s'obtienne par 8a + 4b + 2c + d (code DCB 8421).

Vérifier que la relation a(b + c) = 0 rend exactement compte de ce que les codes de 10, 11, 12, 13, 14, 15 ne seront jamais employés.

En déduire que toute fonction booléenne g(a, b, c, d) *où abcd figure le code 8421 d'un chiffre décimal* peut être réalisée comme $g(a, b, c, d) + \phi\, a(b + c)$.

En déduire l'expression booléenne la plus simple de la fonction $f_6(a, b, c, d)$ = « abcd vaut 6 ou plus ».

15.13 PROBLÈMES

15.13.1 Cellule logique universelle

Une cellule logique réalise la fonction $f(a, b, c, d, e) = ((a \oplus b) \cdot (c \oplus d)) \oplus e$.

En jouant sur l'affectation aux différentes entrées de variables et/ou de constantes, montrer que cette fonction réalise toute fonction de 2 variables.

15.13.2 Croisement virtuel

Au milieu d'un circuit intégré, il faudrait que les connexions (a, b) permutent en (b, a) sans croiser physiquement ces connexions. Montrer qu'on peut le faire à l'aide de 3 *ou exclusifs*.

15.13.3 Fonctions duales et autoduales

Une fonction f est dite autoduale si $f = f^*$.

Montrer que c'est le cas pour la fonction booléenne Maj(x,y,z) – dite encore *vote-taker* ou *majority gate* – dont la sortie s'aligne sur la majorité des entrées.

Généraliser à un nombre d'entrées 2k + 3. Exprimer Maj en tant qu'outil à seuil.

Les fonctions qui ne sont pas autoduales sont duales par paires. Montrer que c'est le cas de toutes les fonctions à 2 variables.

15.13.4

Montrer qu'un additionneur binaire peut être entièrement fait à l'aide de cellules à 3 entrées et 2 sorties : $ad(x, y, z \rightarrow r, s) \equiv \{ r = Maj(x, y, z), s = x \oplus y \oplus z. \}$

Montrer qu'un multiplicateur binaire peut être entièrement fait à l'aide de cellules à 4 entrées et 2 sorties : $mul(a, b, x, y \rightarrow r, s) \equiv \{ r = Maj(x, y, a \cdot b), s = x \oplus y \oplus a \cdot b \}$.

Adapter les deux exercices précédents au cas (1) de nombres en DCB (2) de nombres dans une base quelconque B.

15.13.5

Dans le style de la section 15.8, définir par un graphe binaire bifonctionnel :

- un additionneur binaire complet :
  $ad(x, y, z \rightarrow r, s) \equiv \{ r = Maj(x, y, z), s = x \oplus y \oplus z \}$;
- un multiplicateur binaire complet :
  $mul(a, b, x, y \rightarrow r, s) \equiv \{ r = Maj(x, y, a \cdot b), s = x \oplus y \oplus a \cdot b \}$.

Conclure sur la sensibilité à l'ordre dans lequel les variables sont traitées.

15.14 PISTE DE RÉFLEXION

15.14.1

Adapter aux fonctions booléennes les notions de fonctions croissantes, décroissantes, monotones en x.

En déduire qu'une famille d'outils logiques ne peut se réduire à des *outils à seuil* tels que Maj, mais qu'il faut aussi des *outils à fenêtre* tels que le ou exclusif.

L'existence d'un inverseur suffirait-elle ?

Conclure sur le rôle des poids négatifs ; et, en neuro-sciences, sur l'intérêt des synapses inhibitrices.

15.15 LECTURES

AKERS S.B., « Binary decision diagrams » in *IEEE Trans. on Computers,* vol. C-27, n° 6, juin 1978, pp. 509-516.

BENZAKEN C., « Algorithmes de dualisation d'une fonction booléenne », *RFTI Chiffres*, 1966, pp. 119-128.

BOOLE G. *Les lois de la pensée,* Ed. J. Vrin, 1992, traduction de *The laws of Thought,* 1854.

BRYANT R.E., « Symbolic boolean manipulation with ordered binary decision diagrams » in *ACM Computing Surveys,* sept. 1992, pp. 293-318.

DIAGNE S.B., *Boole (1815-1864), l'oiseau de nuit en plein jour*, Editions Belin, 1989.

FAURE R. et HEURGON E., *Structures ordonnnées et algèbres de Boole*, Collection Programmation, Gauthier-Villars, 1971 (algèbres de Boole et treillis).

FRÉCON L. *Vers des machines intelligentes,* DEA Dimensions Cognitives et Modélisation, Université Lyon II, 2001.

FRÉCON L., *Les bases de connaissances d'ordre 0 comme polynômes booléens,* rapport de recherche, Laboratoire DELIA, INSA Lyon, nov 1993.

KUNTZMANN J., *Algèbre de Boole*, Dunod, 1968.

(Traite des équations booléennes et des fonctions booléennes très indéterminées.)

VALLÉE R.L., *Analyse binaire,* Masson, 1970.

(Curiosité ; dans l'esprit de G. Boole, tente de considérer $\mathbb{B}$ comme sous-ensemble de $\mathbb{R}$; introduit des opérateurs de retard en vue de la prévention des aléas dans le fonctionnement des automates.)

CHAPITRE 16

ALGÈBRE DE KLEENE & AUTOMATES À ÉTATS FINIS

Ce chapitre donne les règles d'équivalences entre

- une classe de *dispositifs* appelés *automates à états finis*, qu'on rencontre en *automatique*, en *architecture de machines*, en *génie logiciel*...
- une classe de *modèles* linguistiques appelés *grammaires de Kleene*, *grammaires linéaires unilatères*, ou encore *grammaires de classe 3*,
- une classe d'expressions de séquences, dites *expressions régulières*, permettant de décrire ou spécifier des *comportements* séquentiels, pouvant comporter des itérations successives et/ou imbriquées, expressions banalisées par l'éditeur QED puis le système unix (cf lex, grep, sed, awk...).

Ces résultats découleront de l'exploitation de gerbiers dûment enrichis.

16.1 NOTION D'AUTOMATE À ÉTATS FINIS

16.1.1 Présentation

DÉFINITION 16.1 – Un ***automate à nombre fini d'états***, ou, par abus de langage, un ***automate à états finis*** est une structure <E, I, S, t, s> à 3 ensembles finis :

- E : alphabet d'entrée
- I : ensemble des états internes
- S : alphabet de sortie

mis en rapport par les 2 fonctions t et s.

DÉFINITION 16.2 – Dans le ***modèle de Mealy***

- la fonction de transition $t : E \otimes I \rightarrow I$, *détermine le prochain état*,
- la fonction de sortie $s : E \otimes I \rightarrow S$ *détermine la sortie.*

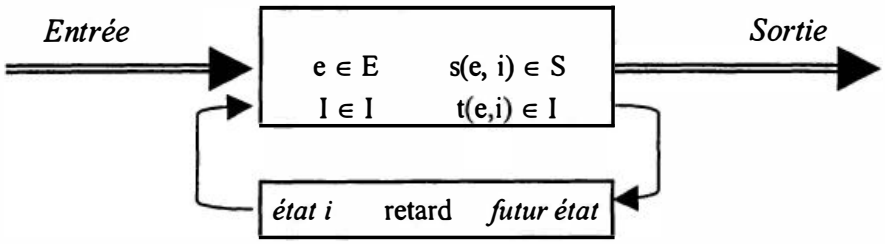


Fig. 16.1 Automate à nombre fini d'états.

DÉFINITION 16.3 – Dans le ***modèle de Moore***

- la fonction de transition $t : E \otimes I \to I$ *détermine le prochain état,*
- la fonction de sortie $s : I \to S$ *détermine la sortie en fonction du seul état.*

Tous les ensembles étant finis, les fonctions t et s peuvent de fait être spécifiées par des tables.

Pour le *modèle de Mealy,* qui fait dépendre la sortie de l'état et de l'entrée, on utilise généralement une table unique T, formée à raison d'une ligne par état interne i, et d'une colonne par entrée possible e ; chaque case ainsi définie comprend un couple (état suivant, sortie) noté i^+/s.

Le *modèle de Moore* juxtapose à une table à 2 entrées définissant le nouvel état $i^+ = T[i, e]$ comme fonction de l'état et de l'entrée, une colonne supplémentaire pour les sorties $s = S[i]$.

DÉFINITION 16.4 – Deux automates ayant même ensemble E et même ensemble S sont dits équivalents si, étant chacun dans son état initial, ils émettent une même séquence de S* chaque fois qu'ils reçoivent une même séquence de E*.

NOTE. C'est une équivalence de comportement et non de fonctionnement, mais on peut dire les fonctionnements compatibles si les comportements sont équivalents.

PROPOSITION 16.1 – Modèle de Mealy et Modèle de Moore sont équivalents :

si $<E, I_1, S, t_1 : E \otimes I_1 \to I_1, s_1 : E \otimes I_1 \to S>$ représente un automate (au sens de Mealy), il existe un $<E, I_2, S, t_2 : E \otimes I_2 \to I_2, s_2 : I_2 \to S>$ représentant un automate équivalent (au sens de Moore), et réciproquement.

16.1.2 Exemple : feux routiers

On veut installer un feu à l'intersection d'une route prioritaire et d'une route secondaire. Celle-ci ne devrait bénéficier d'un cycle que sur un signal D indiquant qu'il y a demande par un piéton, ou un véhicule détecté. On suppose une horloge avec une période de l'ordre de la minute.

On considère comme sorties les éléments convenables de $\{V, J, R\} \otimes \{v, j, r\}$, où v, j, r désignent respectivement les feux vert, jaune et rouge, par une majuscule pour la route principale, une minuscule pour la route secondaire.

L'automate correspondant peut être défini dans le style de Mealy (tab. 16.1) ou dans le style de Moore (tab. 16.2), plus adapté lorsqu'un état correspond à une phase.

Tableau 16.1 Feux routiers (modèle Mealy).

i^+/s	entrées	
états	D′	D
0	0 /Vr	1 /Vr
1	2 /Jr	2 /Jr
2	3 /Rv	2 /Rv
3	0 /Rj	0 /Rj

On dit un ***état stable*** s'il peut être son successeur immédiat.

On constate une *stabilité des états* 0 (en l'absence de demande, route prioritaire passante) et 2 (route secondaire maintenue passante si la demande se maintient). Il y a donc lieu de prévoir que (2) puisse émettre un acquittement de la demande.

Tableau 16.2 Feux routiers (modèle Moore).

	entrées		
états	D'	D	sorties
0	0	1	Vr
1	2	2	Jr
2	3	2	Rv
3	0	0	Rj

16.1.3 Réalisation

Matérielle

Il suffit de pouvoir réaliser

- les 2 fonctions discrètes s et t, ce qui suppose 2 réseaux combinatoires parfois très simples, pas forcément disjoints ;
- une fonction de retard, assurant le renouvellement de l'état interne sans confusion dans le fonctionnement.

Le fonctionnement des *automates synchrones* est cadencé par une horloge biphasée : le retard est assuré par un registre en maître/esclave, dont les entrées sont mises à jour aux phases paires, les sorties aux phases impaires.

Cette fonction d'automate synchrone, repérée depuis longtemps, a été concrétisée chez Texas Instruments, dès les années 70, par un circuit intégré générique comprenant une mémoire morte programmable aux spécifications du client (ce qui revient à câbler la définition logicielle ci-après).

Les *automates asynchrones*, plus performants, sont de conception délicate à cause des *aléas* : l'apparition fugace d'(un code d')état imprévu, du fait de retards inégaux dans les réseaux logiques, peut entraîner une transition et, de là, une séquence erronée.

Logicielle

Le principe en est donné ci-après.

La lecture assure la synchronisation : ce peut être une acquisition physique, un caractère dans un flot, un cas d'un menu...

L'ordre de sortie est similaire.

Modifier i en bloc et en dernier élimine tout problème de confusion entre états.

procédé automate

```
ensembles E, S, I ;
tables t[E, I] de I, s[E, I] de S ;
var e :E, i :I, s :S ;
constante i0 :I ;
    ? initialiser t[,],s[,]?
    i←i0 ;
    indéfiniment faire
        lire e ;
        sortir s[e,i] ;
        i ← t [e,i]
```

fin automate

16.1.4 Graphe de transition

Les tables précédentes mènent à un (multi)graphe de transition :

- à tout état i de l'automate correspond un sommet du graphe de transition,
- [Mealy] à toute case T[e,i] = i^+ / s correspond un arc (i, i+) marqué (e / s),
- [Moore] à tout sommet i est attaché la sortie S[i] correspondante, et toute transition T[e,i] = i^+ est figurée par un arc (i, i^+) marqué e.

L'exemple précédent mène aux graphes de transition 16.2 / 16.3.

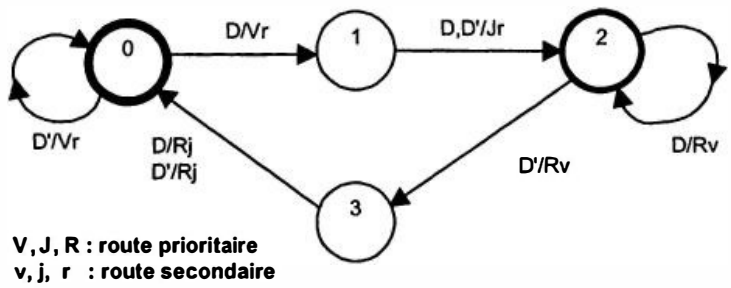


Fig. 16.2 Feux routiers : graphe de transition (Mealy).

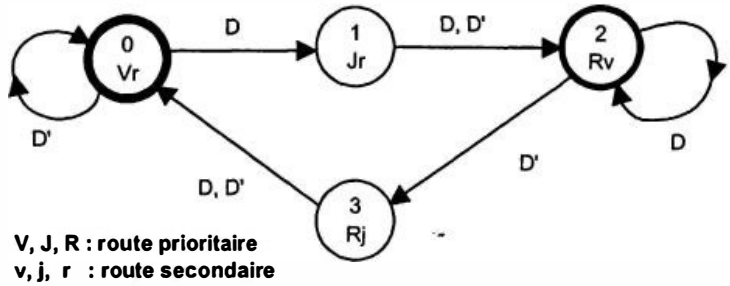


Fig. 16.3 Feux routiers : graphe de transition (Moore).

16.1.5 Principaux comportements

Générateur

Un automate sera considéré comme un ***générateur*** (plus ou moins piloté) s'il est dominé par les sorties, les entrées significatives étant rares.

EXEMPLES. Horloge, automate pilote d'une application, d'un processeur.

Tableau 16.3 Tables d'une horloge pilotée par marche/arrêt ; un bip est donné par l'état instable (2).

état	entrées			sorties
	–	Marche	Arrêt	
0	$\boxed{0}$	1	$\boxed{0}$	–
1	*2*	*2*	0	–
2	*1*	*1*	0	bip

Accepteur, classifieur, analyseur

Considérons un automate dominé par les entrées, et aux sorties plutôt rares. On parle d'***accepteur*** si les sorties significatives sont limitées à acceptation/rejet, de ***classifieur*** si la séquence acceptée est l'objet d'une classification, d'***analyseur*** dans le cas d'une sortie encore plus riche.

EXEMPLE. Analyseur lexical, pouvant reconnaître des séquences comme mots ou comme constantes/ variables/ mots-clés, éventuellement assortis d'un type, d'une valeur ou d'un index.

Traducteur

Un automate sera considéré comme traducteur si les séquences de sortie suivent de près les séquences d'entrée.

16.2 GRAMMAIRES DE KLEENE

Cette section prolonge la section 13.6.

DÉFINITION 16.5 – On considère un vocabulaire V = N + T formé d'un vocabulaire des notions N et d'un vocabulaire des terminaux T, disjoint de N. Convenons que les minuscules latines désignent des éléments de T, les minuscules grecques ceux de N.

*On parle de **grammaire de Kleene**, ou de **grammaire de classe 3** (Chomsky) ou encore de **grammaire linéaire unilatère**, si les règles sont*

- *soit du type $\alpha \to x$ soit du type $\alpha \to x\beta$ pour une **grammaire linéaire [unilatère] gauche**,*
- *soit du type $\alpha \to x$ soit du type $\alpha \to \beta x$ pour une **grammaire linéaire [unilatère] droite**.*

EXEMPLE. Un identificateur de chaîne Basic simplifié peut être ainsi défini par <ι> :

$\iota \to a\sigma \mid b\sigma \mid c\sigma$ $\qquad$ $\sigma \to \$ \mid a\sigma \mid b\sigma \mid c\sigma \mid 1\sigma \mid 2\sigma \mid 3\sigma$

16.2.1 Graphe de transition

Considérons une grammaire du type $\alpha \to x \mid y\beta$.

On peut lui associer un graphe de transition, sur la base

- d'un sommet par notion,
- d'un arc (α, β) marqué y par règle $\alpha \rightarrow y\beta$,
- d'un arc (α, ε) marqué x par règle $\alpha \rightarrow x$, ε marquant ici une bonne fin.

EXEMPLE. Les règles : $\iota \rightarrow a\sigma|\ b\sigma|\ c\sigma$ $\sigma \rightarrow \$|\ a\sigma|\ b\sigma|\ c\sigma|1\sigma|\ 2\sigma|\ 3\sigma$ donnent le graphe de la figure 16.4.

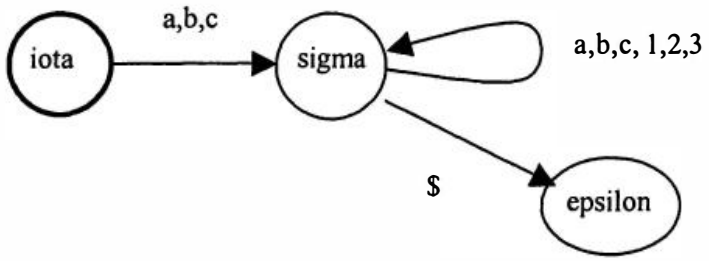


Fig. 16.4 Graphe de transition associé à une grammaire de Kleene.

16.2.2 Equations régissant le langage modélisé

Partant des *paquets de règles* regroupant toutes les règles de grammaire relatives à une même notion (resp. tous les arcs issus d'un même sommet), Schützenberger suggère de leur associer un système d'équations où

- les notions tiennent lieu de variables,
- les terminaux de constantes,
- une somme + se substitue au | des grammaires,
- un produit · explicite la concaténation.

EXEMPLE

$\iota \rightarrow a\sigma|\ b\sigma|\ c\sigma \quad \Rightarrow \quad I = (a + b + c) \cdot S$

$\sigma \rightarrow \$|\ a\sigma|\ b\sigma|\ c\sigma|1\sigma|\ 2\sigma|\ 3\sigma \quad \Rightarrow \quad S = (a + b + c + 1 + 2 + 3) \cdot S + \$$

Résoudre de tels systèmes dans une structure algébrique adéquate, comportant au moins une concaténation (pour bâtir des séquences) et une agrégation (pour les réunir), associerait à chaque notion (et notamment la notion initiale) l'expression algébrique du langage correspondant.

16.3 ALGÈBRE DE KLEENE

16.3.1 Dioïde de base

Considèrons le dioïde $A^{+\cdot} = \langle A, \omega, \varepsilon, +, \cdot \rangle$ formé de

- un sous-ensemble A fini de générateurs, dit alphabet,
- une opération +, associative, *commutative et idempotente,*
- un élément ω neutre pour +,
- une opération ·, associative, *non commutative*, distributive sur la première, d'élément neutre ε et d'élément absorbant ω.

On peut voir cette structure comme base d'une *algèbre des chaînes*

- d'alphabet A,
- + jouant le rôle d'union et · celui de concaténation,
- ε jouant le rôle de la *chaîne vide* et ω celui d'une *chaîne impossible.*

16.3.2 Expansion de Kleene

Considérons un système d'équations du genre :

$$I = (a + b + c) \cdot S$$
$$S = (a + b + c + 1 + 2 + 3) \cdot S + \$$$

où I et S sont maintenant des ***variables linguistiques***. Visiblement, il a un sens dans le dioïde introduit.

Pour le résoudre, éliminons dans un premier temps les variables qui ne dépendent pas explicitement d'elles-mêmes. Le système se réduit alors à des équations du genre $Z = f(Z)$.

Compte tenu des grammaires unilatères dont nous sommes partis, les équations résiduelles seront d'un des types :

$$Z = a + b \cdot Z \quad \text{ou} \quad Z = a + Z \cdot b$$

mais pas des deux.

Soit donc à résoudre dans $A^{+\cdot}$ le système $Z = a + b \cdot Z$. Décidons d'atteindre la solution Z comme limite formelle d'une suite :

$$Z_0 = a, \qquad Z_n = a + b \cdot Z_{n-1}$$

Visiblement,

$Z_0 = a,$
$Z_1 = a + b \cdot Z_0 = a + b \cdot a = (\varepsilon + b)\, a$
$Z_2 = a + b \cdot Z_1 = a + b \cdot (a + b \cdot a) = a + b \cdot a + b \cdot b \cdot a = (\varepsilon + b + b^2) \cdot a$
$Z_3 = a + b \cdot Z_2 = a + b \cdot (a + b \cdot a + b \cdot b \cdot a) = a + b \cdot a + b \cdot b \cdot a + b \cdot b \cdot b \cdot a$
$= (\varepsilon + b + b^2 + b^3) \cdot a$

Par récurrence, on vérifie que l'on a bien : $Z_n = a \cdot \sum_{i=0}^{n} b^i$

On pose maintenant :

DÉFINITION 16.6 – Dans un dioïde, l'opération dite ***expansion de Kleene***, notée *, est la somme des puissances successives de 0 à l'infini. De même $^{\oplus}$ dénote la somme des puissances successives de 1 à l'infini.

Soit : $x^* \equiv \sum_{i=0}^{\infty} x^i, \quad x^{\oplus} \equiv \sum_{i=1}^{\infty} x^i$

PROPOSITIONS

16.2 – L'expansion de Kleene vérifie les relations :

$x^* = \varepsilon + x^{\oplus}$	$x^* \cdot x^* = x^*$
$\varepsilon^* = \varepsilon$	$x^* \cdot x^{\oplus} = x^{\oplus} \cdot x^* = x^{\oplus}$
$(x + \varepsilon)^* = x^*$	$(x^*)^* = x^* = (x^{\oplus})^* = (x^*)^{\oplus}$
$x \cdot x^* = x^* \cdot x = x^{\oplus}$	$(x^{\oplus})^{\oplus} = x^{\oplus}$

16.3 – *$Z = b^* \cdot a$ est la solution de $Z = a + b \cdot Z$,*

PREUVE. $a + b \cdot Z = a + b \cdot (b^* \cdot a) = a + b^{\oplus} \cdot a = (\varepsilon + b^{\oplus}) \cdot a = b^* \cdot a = Z$. ■

De même :

16.4 – *La solution de $Z = a + Z \cdot b$ est $Z = a \cdot b^*$.*

EXEMPLE. La solution de $I = (a+b+c) \cdot S$

$S = (a+b+c+1+2+3) \cdot S + \$$

est donnée par $S = (a+b+c+1+2+3)^* \$$

d'où $I = (a + b + c) \cdot (a + b + c + 1 + 2 + 3)^* \$$

ce qu'on peut noter dans un style cobolisant :

$$I = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \left[\begin{Bmatrix} a \\ b \\ c \\ 1 \\ 2 \\ 3 \end{Bmatrix} \ldots \right] \$$$

NOTE. Dans un système linéaire traditionnel, on écrirait :

$Z = a + Z \cdot b \rightarrow Z \cdot (1 - b) = a \rightarrow Z = a/(1 - b) = a \cdot (1 + b + b^2 + b^3 + b^4 \ldots)$ quand $|b|<1$; ainsi, b^* est l'analogue, *dans des structures plus pauvres*, de $(1-b)^{-1}$.

16.3.3 Expressions régulières

DÉFINITION 16.7 – On appelle ***expression régulière*** bâtie sur l'alphabet A :

- ε, l'expression vide, et ω, l'expression impossible,
- tout élément de A,
- toute expansion E^*, tout reflet $\tilde{E}$ d'une expression régulière E,
- toute union, différence ensembliste, intersection ou concaténation d'expressions régulières E_1 et E_2, notées respectivement $E_1 \cup E_2$, $E_1 - E_2$ $E_1 \cap E_2$, $E_1 \cdot E_2$.

16.3.4 Reflet d'une expression régulière

DÉFINITION 16.8 – Le ***reflet d'une séquence*** a est une image ã formée de la suite des mêmes éléments pris dans l'ordre inverse – comme dans un miroir.

$$\forall x \in A,\ \tilde{x} = x$$

$$\forall x \in A, y \in A^* : z = x \cdot y \rightarrow \tilde{z} = \tilde{y} \cdot \tilde{x}$$

ainsi : $\widetilde{a \cdot b \cdot c \cdot d \cdot e} = e \cdot d \cdot c \cdot b \cdot a$; au-delà :

$$\forall x, y \in A^* : \widetilde{x + y} = \tilde{x} + \tilde{y},$$

$$\forall x \in A^* : \tilde{x^*} = (\tilde{x})^*$$

Reflets et parenthèses

Algol 68 voit *toute parenthèse comme reflet de la parenthèse appariée,* en étendant cette idée à tout *mot-clé* à statut parenthétique.

Tableau 16.4 Système de parenthèses genre Algol 68.

parenthèse gauche	parenthèse droite
(	)
[	]
{	}
case	esac
do	od
fortran	nartrof
if	fi

Palindromes

DÉFINITION 16.9 – Les ***palindromes*** sont les points fixes de la réflexion, mots qui sont leur propre reflet [dans un miroir].

EXEMPLES. y, ara, gag, non, sas, ses, sis, sus, elle, kayak, rotor, taxât, selles …

16.4 PROPRIÉTÉ CENTRALE

PROPOSITION 16.5 – *Le dioïde A^{+*} = <A, ω, ε, +, · > est stable pour les somme, concaténation, expansion, intersection, différence et complément à A^{+*}, et réflexion.*

On peut donc le noter : A^{+*} = <A, ω, ε,+, ·,* , ∧, –, ~>.

Le dioïde précédemment défini devient support de l'***algèbre de Kleene***, en ce qu'il est le domaine nécessaire et suffisant d'évaluation des expressions régulières bâties sur le même alphabet.

16.4.1 Exemple : restructuration de programmes

On considère l'***organigramme*** donné figure 16.5, et on recherche le ***programme structuré*** réalisant le même traitement.

On convient d'appeler Tn le traitement réalisé à l'entrée de l'étape n (action, convergence ou divergence).

La concaténation, notée « · », code facilement les séquences.

Les choix booléens seront codés par une variable minuscule correspondante dans la forme convenable.

Enfin, le traitement associé à un point de convergence sera la somme des traitements effectués sur chacune des branches qui y arrive.

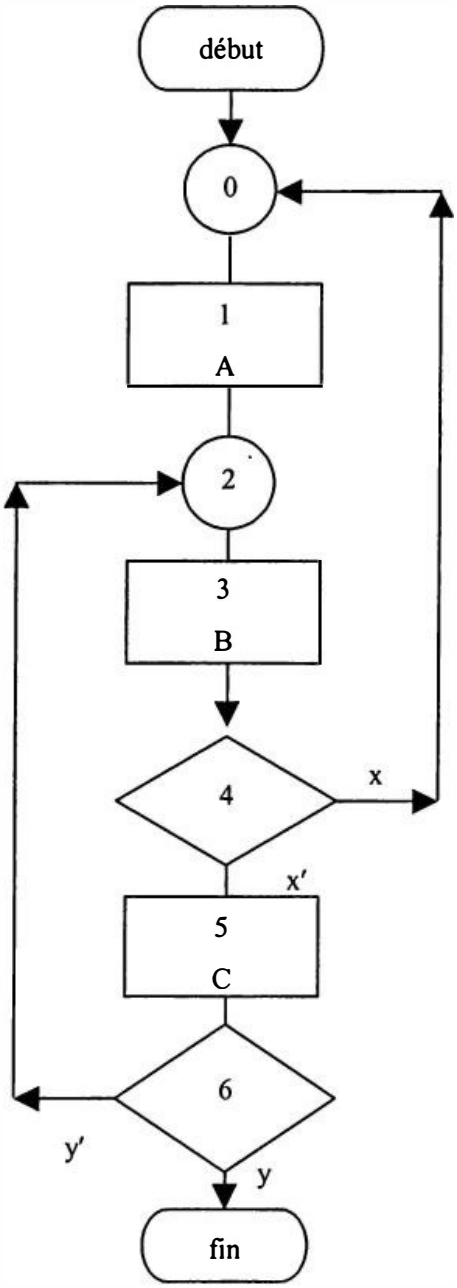


Fig. 16.5 Programme à structurer.

On obtient d'abord les équations locales :

T0 = début + T4·x
T1 = T0
T2 = T1·A+T6·y'
T3 = T2
T4 = T3·B
T5 = T4·x'
T6 = T5·C
fin = T6y

En éliminant les variables utilisées une seule fois il vient :	T0 = début + T4·x T2 = T0·A+T6·y′ T4 = T2·B T6 = T4·x′·C fin = T6·y
puis en éliminant T4	T0 = début + T2·B·x T2 = T0·A+T6·y′ T6 = T2·B·x′C fin = T6·y
puis T6	T0 = début + T2·B·x T2 = T0·A+T2·B·x′·C·y′ fin = T2·B·x′·C·y
puis T0	T2 = début·A+ T2·B·x·A+T2·B·x′·C·y′ fin = T2·B·x′·C·y
d'où finalement	T2 = début·A·(B·x·A+B·x′·C·y′)* fin = début·A·(B·x·A+B·x′·C·y′)*·B·x′·C·y

Le problème est maintenant de faire apparaître dans cette *formule décrivant l'effet du programme*, les formes typiques de la programmation structurée.

Tableau 16. 5 Schémas de programmes et expressions formelles.

schéma de programme	expression régulière associée
A ; B ; C	A·B·C
si x alors A sinon B	x·A+x′·B
si x alors A	(x·A+x′)
tantque x faire S	(x·S)*x′
répéter S jusqu'à x	S(x′·S)*x

La somme centrale n'a pas la forme adéquate, mais une *suite quelconque de a et de b* peut être réécrite comme *suite de « suite de a suivis d'un b », suivie éventuellement d'une dernière suite de a :*

$$(a + b)^* = (a^*b)^*a^*$$

fin = début·A·(B·x·A+Bx′Cy′)*·Bx′·C·y peut donc se réécrire

fin = début·A·((B·x·A)*B·x′·C·y′)*(B·x·A)*·B·x′·C·y
= début A·((B·x·A)*B·x′·C·y′)*(B·x·A)*B·x′·C·y

soit en posant S = (B·x·A)*B·x′·C

fin = début·A·(S·y′)*S·y = début·A·S·(y′·S)*y

*début · A · S · (y′ · S)*y* correspond au schéma *début A ; répéter S jusqu'à y ; fin*

S s'explicite maintenant en notant que :

$$(ba)^*b = b + bab + babab + \ldots = b(ab)^*$$

Ainsi S = (BxA)*Bx′C = B(xAB)*x′C, ce qui correspond au schéma
[B ; tantque x [A ; B] ; C].

Finalement, *le programme initial est équivalent au programme de la figure 16.6.*

VÉRIFICATION. On vérifie pas à pas que des *expressions de traitements* du programme 1, simples comme ABx'Cy ou plus complexes comme

$$(ABx)^{12}A(Bx'Cy')^{3}B(ABx)^{5}ABx'Cy$$

décrivent chacune un comportement du programme 2.

```
début
A
répéter
    B
    tantque x faire
        A
        B
    C
jusqu'à y
fin
```

Fig. 16.6 Programme structuré obtenu.

LIMITES. Le procédé ne traite que les programmes itératifs et ne peut rendre compte des interdépendances entre ces itérations : c'est pourquoi il ne peut être appliqué directement qu'aux fausses récursivités ; les récursivités intrinsèques exigeraient au moins de recourir à des expressions de classe 2.

- Soit une action à récursion terminale :

  écrire (phrase) ≡ [si non(vide(phrase)) alors
  [écrireMot(premier(phrase)) ; écrire(reste(phrase))]]

  on peut la formaliser par : e = v'me+v soit e = (v'm)*v correspondant aussi à

  écrire(phrase) ≡ [tantque(non(vide(phrase))
  [écrireMot(premier(phrase)) ; phrase ← reste(phrase)]
  ]

  en posant que m symbolisant « toute action précédant le rappel de e » comprend aussi bien l'écriture du mot de tête que la réduction de la phrase restant à écrire. Dans ce cas il y a *récursivité de surface.*

- Soit une fonction récursive, dont le rappel implique pré- et post-traitement.

  Le schéma f () ≡ [si v alors g·f·h sinon j]
  se transcrit formellement $f = (v \cdot g \cdot f \cdot h + v' \cdot j)$
  dont la solution $f = \Sigma_k f_k$ avec $f_k = (v \cdot g)^k \cdot v' j \cdot h^k$
  relève des langages de classe 2 (qui rendent compte des structures de parenthèse), *au-delà des expressions régulières*, et on parlera de ***récursivité intrinsèque.***

  Ainsi, la fonction : fact(n) ≡ [si n = 0 alors 1 sinon n·fact(n–1)]
  comporte en prétraitement du rappel la réduction de l'argument, et en post-traitement le produit, *effectué au retour* de l'appel récursif.
  Dans un procédé aligné sur ces langages à parenthèses, fact(n) aurait comme « image itérative brute » quelque chose comme

  fact(n) ≡ [tantque n>0 [empiler n ; n ←n–1] « image des prétraitements
  fact ←1 « cas central
  tantque non(vide(pile)) [fact ←fact·sommet]
  « image des post-traitements, si lire le sommet le dépile
  ]

  menant par une suite d'améliorations ponctuelles[1] à

  fact(n) ≡ [fact ← 1 ; pour i ← 2 à n [fact ← fact·i]].

[1] Voir aussi chapitre 6.

16.5 DES EXPRESSIONS RÉGULIÈRES AUX GRAPHES DE TRANSITION

Automates à états finis et grammaires de Kleene partagent des graphes de transition, correspondant à des expressions régulières ; ainsi passe-t-on du fonctionnement d'un dispositif ou d'une grammaire/modèle à la description de comportements observables, ce qui est très utile à la compréhension d'un existant.

Symétriquement, peut-on passer d'une expression régulière caractérisant un comportement recherché

- à une grammaire qui en donne une analyse, si sommaire soit-elle ?
- à un automate à états finis pouvant assurer un traitement recherché (génération, détection, traduction) ?

Le point essentiel sera d'associer à une expression régulière précise un graphe de transition.

16.5.1 Dérivée d'une expression régulière

Supposons une expression régulière du genre E = aF + bG + cH ; il est facile de lui associer un (fragment de) graphe (de transition) : on assigne les expressions E, F, G, H à autant de sommets, et on marque respectivement a, b, c les arcs (E, F), (E, G), (E, H) (fig. 16.7).

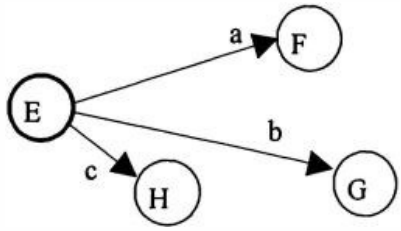


Fig. 16.7 Graphe de transition associé à $E = a{\cdot}F + b{\cdot}G + cH$.

On peut considérer F, G, H comme *dérivés* de E relativement à a, b, c.

Pour réaliser la même opération sur une expression régulière quelconque, nous utilisons une notion générale de *dérivée au sens de Booth* d'une expression régulière par rapport à un terminal.

DÉFINITION 16.10 – On appelle ***dérivée d'une expression régulière E par rapport à un terminal t*** de l'alphabet A de E, une fonction $D_t(E)$ ainsi définie :

$$\begin{array}{lll} \forall t, x \in A & D_t(x) \equiv & (\, t = x \rightarrow \varepsilon \mid \omega \,) \\ \forall t, x \in A, E \in A^{+*} & D_t(x{\cdot}E) \equiv & (\, t = x \rightarrow E \mid x = \varepsilon \rightarrow D_t(E) \mid \omega \,) \\ \forall t \in A, E,F \in A^{+*} & D_t(E + F) \equiv & D_t(E) + D_t(F) \end{array}$$

PROPOSITION 16.6 – La dérivée d'une expression au sens de Booth satisfait les propriétés suivantes :

$$\begin{array}{ll} \forall t \in A, E,F \in A^{+*} & D_t(E{\cdot}F) \equiv (\, E = \varepsilon + E_1 \rightarrow D_t(E_1){\cdot}F + D_t(F) \mid D_t(E){\cdot}F \,) \\ \forall t \in A, E \in A^{+*} & D_t(E^*) \equiv D_t(E){\cdot}E^* \\ \forall t \in A, E,F \in A^{+*} & D_t(E^*{\cdot}F) \equiv D_t(F) + D_t(E){\cdot}E^*{\cdot}F \end{array}$$

REMARQUE. $D_t(E^*)$ s'obtient comme $D_t(\varepsilon + E{\cdot}E^*)$.

16.5.2 Construction d'un graphe de transition

Soit à construire le graphe de transition associé à une expression E bâtie sur A.

Le graphe est formé

- d'un sommet par expression nouvelle à traiter,
- d'un arc (E, F) marqué x *ssi* $D_x(E) = F$.

Pour cela, on forme progressivement la table des transitions à raison

- d'une première ligne pour l'expression initiale,
- d'une colonne par élément significatif de A,
- dans chaque case, la dérivée de l'expression de la ligne par le terminal de la colonne,
- d'une nouvelle ligne par expression *nouvelle* rencontrée.

Une expression réduite à ε est le signe de bonne fin, une expression contenant ε dénote une fin possible, une expression réduite à ω dénote une impasse ou erreur.

Attention ! le graphe obtenu comporte un nombre minimal de sommets ssi on ne crée pas de lignes inutiles,

i.e. si la détection de l'équivalence entre expressions est parfaite

16.5.3 Exemple

Soit à traiter E = (ab*a)*ab. On dresse progressivement la table 16.6, en calculant successivement :

$D_a(E) = D_a(ab + ab^*a(ab^*a)^*ab) = b + b^*a(ab^*a)^*ab = F$

$D_b(E) = D_b(ab + ab^*a(ab^*a)^*ab) = \omega + \omega = \omega$

$D_a(F) = D_a(b + b^*a(ab^*a)^*ab) = D_a(b^*a(ab^*a)^*ab) = D_a(a(ab^*a)^*ab + bb^*a(ab^*a)^*ab) = (ab^*a)^*ab = E \ldots$

Si on désire que l'automate soit utilisable répétitivement, il faut ajouter

- un état pour ω (collecte des erreurs) ;
- un symbole terminateur indiquant qu'une séquence est finie, ce qui permet de redémarrer l'analyse pour une nouvelle expression, et de lever certaines ambiguités.

Tableau 16.6 Table de transition associée à (ab*a)*ab.

expression	nom	a	b	#
(ab*a)*ab	E	F	Z/ko	E/ko
b + b*a(ab*a)*ab	F	E	G	E/ko
ε + b*a(ab*a)*ab	G	E	H	E/ok
b*a(ab*a)ab	H	E	H	E/ko
ω	Z	Z	Z	E

16.5.4 Graphes de transition et grammaires

A tout graphe de transition correspond une grammaire, sur la base d'une notion par état : ici, au graphe de la figure 16.8 correspond :

$E \rightarrow aF$
$F \rightarrow aE \mid bG$
$G \rightarrow aE \mid bH \mid \varepsilon$
$H \rightarrow aE \mid bH$

Ambiguités

Dans cette grammaire, la notion G est ambiguë, du fait qu'une transition E-G correspond à la fois à une expression régulière du type cherché, et au début d'une expression régulière strictement plus longue mais relevant de la même forme.

EXEMPLE. ab est à la fois forme complète, et le début de formes $ab^{n+1}a(ab^*a)^*ab$ relevant du même modèle.

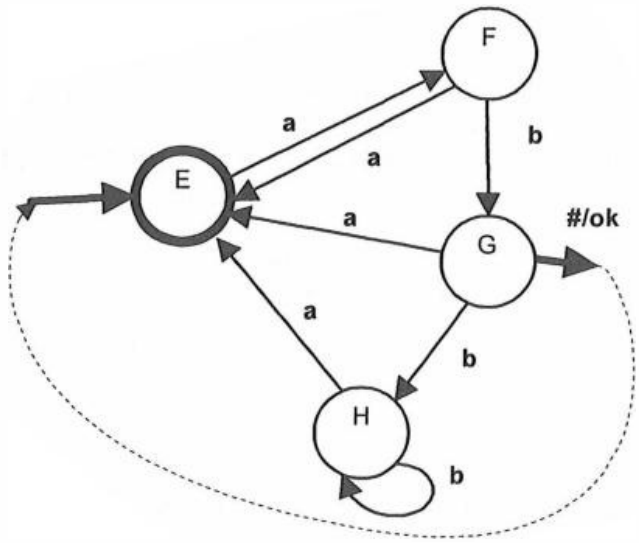


Fig. 16.8 Graphe de transition associé à (ab*a)*ab.

DÉFINITION 16.11 – On dira qu'une forme licite complète est une ***forme close*** ssi *elle n'est pas préfixe d'une forme complète plus longue.*

On peut rendre close toute forme licite complète en introduisant un terminateur (blanc, ponctuation, délimiteur quelconque).

C'est ce qui permet de discriminer {se, ses, session, sessions}.

C'est aussi ce qui permet de distinguer une constante entière d'une constante réelle, la constante entière étant le plus souvent une suite de chiffres *suivie de rien d'autre* : ni chiffre, ni partie fractionnaire, ni facteur de cadrage.

Dans tous ces cas, la décision est rapide car non seulement les séquences plus longues sont finies, mais la suite se distingue dès le premier caractère.

Généralement, un graphe de transition ou une grammaire sont ambiguës lorsqu'un terminal autorise deux transitions distinctes, comme dans :

$$P \rightarrow \mathbf{a}X \mid \mathbf{a}Y$$
$$X \rightarrow aY \mid b \mid cY$$
$$Y \rightarrow bP \mid cX$$

L'idée est alors de *factoriser le terminal commun*, menant à une notion nouvelle cumulant les propriétés des notions en concurrence, et de propager cette modification jusqu'à stabilisation. Posons ici Q = X + Y, et donnons à Q comme développement l'union des développements de X et Y :

$$P \rightarrow aQ$$
$$Q \rightarrow aY \mid b \mid \mathbf{c}Y \mid bP \mid \mathbf{c}X$$
$$X \rightarrow aY \mid b \mid cY$$
$$Y \rightarrow bP \mid cX$$

Une nouvelle ambiguïté apparaît pour le terminal c, mais la double destination étant encore le couple (X, Y), il suffit de prendre Q comme successeur unique.

D'où, en posant de plus R = P + ε, la nouvelle formulation :

$$\begin{aligned} P &\rightarrow aQ \\ Q &\rightarrow aY \mid bR \mid cQ \\ R &\rightarrow aQ \mid \varepsilon \\ X &\rightarrow aY \mid cY \mid b \\ Y &\rightarrow bP \mid cX \end{aligned}$$

16.5.5 Graphes de transition et automates

A tout *graphe de transition* correspond un *automate* de même *table de transition*. En fait, cet automate peut être enrichi selon l'action attendue :

- générateur pur : les terminaux désignent des sorties ;
- accepteur : les terminaux sont des entrées, les sorties sont des diagnostics plus ou moins évolués, au moins oui/non ;
- traducteur : les terminaux sont a priori des couples (entrée, sortie).

De plus, on devra prendre en compte le fait que l'action est faite une seule fois ou répétitivement : il faut alors prévoir reprise sur succès /reprise sur erreur.

Déterminisme, non-déterminisme

DÉFINITION 16.12 – Un automate sera dit ***automate non déterministe*** si, dans un état donné, un même terminal autorise deux transitions distinctes.

Cette question est considérée comme

- peu importante, s'il s'agit d'un module réalisant un générateur,
- critique dans tous les autres cas, que le non-déterminisme soit cause d'erreurs de comportement (classification, analyse, traduction) ou de fonctionnement, au niveau de la technologie.

On a à ce sujet :

PROPOSITION 16.7 – A tout automate non déterministe à k états correspond un automate déterministe à 2^k états au plus acceptant ou engendrant les mêmes séquences.

Partant d'un automate non-déterministe A, l'idée générale est de bâtir un automate déterministe A′ dont chaque état correspond à une *collection d'états* de A atteinte à partir de l'état initial i_0. Dans A′, la première ligne de la nouvelle table est relative au singleton $\{i_0\}$, qu'on développe en fonction des combinaisons atteintes.

Le processus converge et élimine les ambiguïtés, à condition d'admettre des *sorties retardées*, correspondant au nombre d'entrées successives nécessaires pour lever l'ambiguïté. De ce fait, il peut y avoir des suites de sorties neutres suivies de séquences émises en une fois.

Soit en effet la table de transition 16.7.

Nous y relevons 2 ambiguïtés dont l'effet global est inconnu.

Construisons une nouvelle table sur la base indiquée. Les lignes sont fusionnées avec (sauf problème de sécurité) priorité aux suites positives : dans un état ambigu,

une séquence acceptée au titre de l'une quelconque des possibilités est globalement acceptée ; elle ne peut être refusée que si elle l'est pour chaque possibilité.

Comme 0 menait à 0, 1+3, et 2 respectivement, {0} mène à {0}, {1, 3} et {2}.

Comme 1 menait à 2, 2 et 3 respectivement, et 3 à 1+ 2, 2 et 2 respectivement, {1, 3} mène à {2}∪{1,2} = {1,2}, {2} et {2, 3} respectivement...

Tableau 16.7 Table de transition non déterministe.

entrée / **état**	**a**	**b**	**c**	**$**
0	0	*1, 3*	2	arrêt/non !
1	2	2	3	arrêt/non !
2	3	arrêt/non !	3	arrêt/oui
3	*1, 2*	2	2	arrêt/non !

En cas de conflit, les messages négatifs sont abandonnés au profit de la ou des autres solutions.

Tableau 16.8 Table de transition déterministe dérivée de 16.7.

	entrées			
état	**a**	**b**	**c**	**$**
{0}	{0}	{1, 3}	{2}	arrêt/non !
{1, 3}	{1, 2}	{2}	{2, 3}	arrêt/non !
{2}	{3}	arrêt/non !	{3}	arrêt/oui
{1, 2}	{2, 3}	{2}	{3}	arrêt/oui
{2, 3}	{1, 2, 3}	{2}	{2,3}	arrêt/oui
{3}	{1, 2}	{2}	{2}	arrêt/non !
{1, 2, 3}	{1, 2, 3}	{2}	{2, 3}	arrêt/oui

Dans le nouvel automate, les ambiguïtés ont disparu, mais le nombre d'états est de 7 ($4 \leq 7 \leq 16$).

L'ancien état 1 n'apparaît plus seul, et devient inutile. Il y a maintenant 4 états chargés de lever des ambiguïtés diverses de l'ancien automate. Celles-ci viennent des entrées a, à un moindre degré c ; une entrée b est au contraire très sélective.

16.6 SYNOPTIQUE

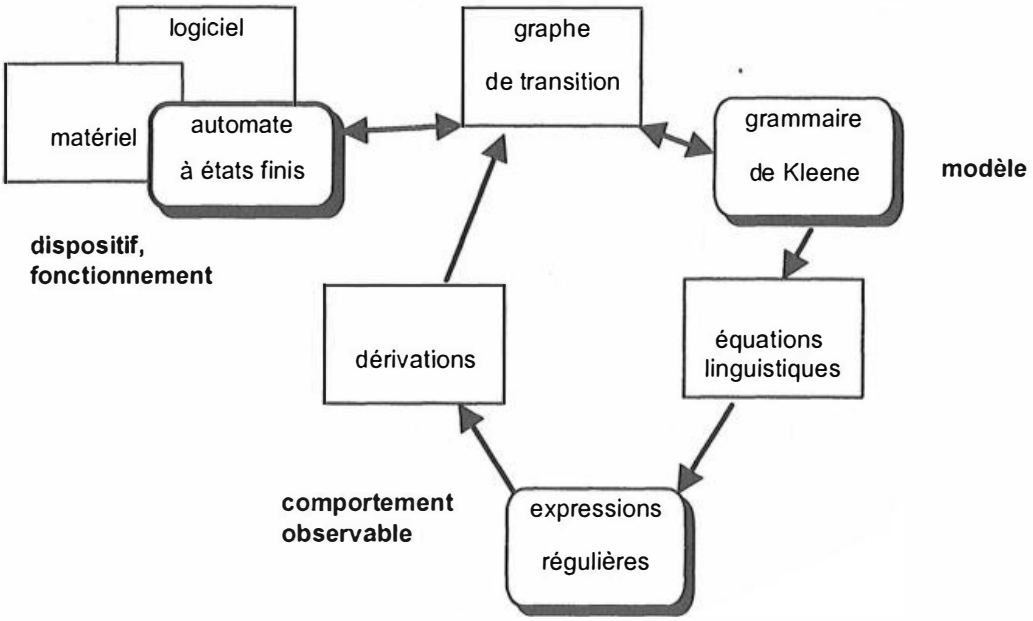


Fig. 16.9 Automates ↔ grammaires ↔ expressions régulières.

16.7 EXERCICES

16.7.0

Définir un automate qui repère toute séquence bâtie sur E = {a,b,c} et se terminant en ab^2c^3.

16.7.1

Modéliser en termes d'automate à 2 états, le comportement d'un chat qui

- s'il ne voit *pas de souris*, en *cherche* une,
- s'il voit une *souris*, (l'attrape et) la *mange* jusqu'à ce qu'il n'en reste plus trace.

16.7.2

Modéliser en termes d'automate le comportement minute par minute d'un individu qui

- répond si on l'appelle plus de 3 fois en 5 minutes,
- vient si on l'appelle une 5e fois dans les 3 minutes qui suivent.

16.7.3

On envisage le codage décimal de textes, par un système à longueur variable qui garantit que 55% des signes d'un texte français soit codé par 1 chiffre décimal, avec en moyenne 1,45 chiffre par signe.

EXEMPLE.

L e p r e s b y t è r e n ' a r i e n p e r d u d e s o n c h a r m e n i l e j a r d i n ...

631072611381855161106020614160072161656206510364600708226171106040631083461654

Concevoir l'automate transcrivant en texte une chaîne chiffrée.

Ce code a-t-il des propriétés statistiques le rendant facile à casser ?

Tableau 16.9 Codage décimal de textes.

suffixe préfixe	0	1	2	3	4	5
rien	blanc	e	a	s	i	t
6	n	r	u	l	o	d
7	c	m	p	v	q	g
8	f	b	h	j	x	y
9	z	k	.	-	(	)

16.7.4

Queneau a proposé en 1937 de simplifier l'orthographe : « on pourrait employer l'alphabet :

a, â, b, d, e, é, è, ê, f, g (toujours dur*), i, j, k, l, m, n, o, ô, p, r, s* (sifflant comme ç ou ss), *t (dur), u, v, y, z, ch, gn, ou, an, in, on* en observant cette règle que toute lettre se prononce, et sans jamais changer de valeur quelque soit sa position. Mézalor, mézalor, késkon obtyin ? Sa devyin incrouayab, pazordinèr, ranversan, sa vouzaalor indsé droldaspé dontonrevyin pa. On lrekonê pudutou, lfransê, amésa pudutou, sa vou pran toudinkou unalur ninversansabarb, sé stupéfian. Avrédir, sémêm maran...»

Concevoir l'automate transcrivant un texte usuel en texte écrit « à la Queneau ».

16.7.5

On envisage le codage duodécimal de textes sur la base de la table ci-après.

Pourquoi w, 0 et 1 n'ont-ils pas de code ?

Tableau 16.10 Codage duodécimal de textes

suffixe préfixe	0	1	2	3	4	5
*	blanc	e	a	s	i	t
#	n	r	u	l	o	d
6	c	m	p	v	q	g
7	f	b	h	j	x	y
8	z	k	.	-	2	3
9	4	5	6	7	8	9

Les ensembles préfixe et suffixe étant disjoints, le préfixe ne sera nécessaire que lorsqu'il change, de telle façon que v o t r e d o n

ait pour code complet 63#4*5#1*1*0#5#4#0

et pour code compact 63#4*5#1*10#540.

Concevoir l'automate transcrivant en texte une chaîne chiffrée pouvant mélanger codes complets et codes abrégés.

Ce code a-t-il des propriétés statistiques le rendant facile à casser ?

Concevoir l'automate de chiffrage d'un texte, obéissant de plus à une commande c, activée lorsqu'on veut un code compact.

Recommandations pour :

- une transmission à haut débit sur une ligne sûre,
- une transmission sur une ligne mauvaise,
- une transmission sur une ligne de bonne qualité, mais avec risque de dérivation sauvage.

Peut-on améliorer l'efficacité de la factorisation des préfixes ?

16.7.7 Trainspotting

Bâtir sur l'alphabet {l, w} (l pour locomotive, w pour wagon) les expressions régulières décrivant

- une rame de wagons R,
- un train T,
- une rame de locomotives haut-le-pied H.

Concevoir un automate pouvant distinguer les trois cas.

16.8 PROBLÈMES

16.8.0 Séquenceur de processeur

Définir sommairement, sur la base d'un état par action, un automate pouvant effectuer le cycle suivant :

- amener dans un registre RI l'instruction désignée par le registre P,
- amener dans un registre AQ l'opérande désigné par adr(RI), la partie adresse de RI,
- exécuter l'opération désignée par op(RI), partie opération de RI,
- incrémenter P et recommencer.

Enrichir cet automate :

- quand doit-on sauter l'étape 2 ?
- caractériser l'exécution d'un arrêt,
- variante du cycle pour une instruction de branchement.

16.8.1 Analyse lexicale

Dans le langage de programmation de votre choix :

- repérer la syntaxe de la constante réelle,
- donner la constante réelle sous la forme d'une expression régulière,
- bâtir l'automate de reconnaissance d'une telle constante.

16.8.2 Automate classifieur

On considère les 3 expressions régulières E = a*(bc)*ca, F = (bab+cc)*aba, G = (cab)*bac.

Combien faut-il d'états pour reconnaître E isolément ? F isolément ? G isolément ?

Adapter la méthode de dérivation au cas où on voudrait tenter de reconnaître *en une fois* si la séquence reçue est (E),(F),(G), ou erronée.

Y a-t-il des cas mixtes ?

Synthèse de l'automate de classification, comparaison de sa complexité à celle des automates antérieurs.

16.8.3 Ascenseur

Soit un immeuble à 5 niveaux (2 extrêmes et 3 ordinaires), à munir d'un ascenseur. Définir l'automate de commande du moteur, ayant pour entrées externes les boutons de destination C0, C1, C2, C3, C4 de la cabine, et les boutons d'appel par niveau A0, A1, A2, A3, A4. Supposer d'abord que l'ascenseur ne répond à un appel que s'il est inactif.

16.8.4 Détecteur de rotation

Dans un système de ventilation de mine ou de tunnel, on désire savoir si un certain ventilateur tourne spontanément – et dans quel sens – lorsqu'il n'est pas commandé. Pour cela, il porte un disque comportant k anneaux concentriques divisés en secteurs soit blancs soit noirs. Une barrette fixe fournit à ces anneaux un mince éclairage radial, et porte une cellule photoélectrique par anneau, donnant par réflexion l'état de son anneau sous l'éclairage.

On suppose k = 4. Définir l'automate détecteur de rotation selon qu'on utilise de secteur en secteur une alternance N/B guidée par

- un code Johnson à 2k = 8 secteurs {0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000},
- un code Gray à $2^k = 16$ secteurs {0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000}.

Que se passe-t-il aux frontières de secteur, qui justifie ces codes par rapport à l'énumération classique ? Quel code vous semble le plus adapté ?

16.9 PISTES DE RÉFLEXION

16.9.1 Code Baudot

Réviser le principe du code Baudot de transmission alphanumérique, avec des *codes de 5 bits.*

Montrer que ce système, poussé à l'extrême, ne permet pas de coder des alphabets de plus de 256 caractères.

16.9.2 Codage du cadran téléphonique

On suppose un texte codé suivant les conventions du cadran téléphonique.

Tableau 16.11 Codage au cadran.

1	2	3	4	5	6	7	8	9	0
blanc	a	d	g	j	m	p	t	w	o
	b	e	h	k	n	r	u	x	q
	c	f	i	l		s	v	y	z

Ce code est ambigu. Sachant cependant que le texte est en français, vérifier que vous arrivez à décoder :

741808718363013462624312770783018614283281

Définir en détail l'automate réalisant systématiquement la même traduction.

Le retard moyen entre entrée et sortie est-il important ? pourquoi ?

16.9.3

Queneau a proposé de simplifier l'orthographe à 2 niveaux. D'abord «*on pourrait employer l'alphabet :* a, â, b, d, e, é, è, ê, f, g (toujours dur), i, j, k, l, m, n, o, ô, p, r, s (toujours sifflant comme ç ou ss), t, u, v, y, z, ch, gn, ou, an, in, on, *en observant cette règle que toute lettre se prononce, et sans jamais changer de valeur quelque soit sa position.* Mézalor, mézalor, késkon obtyin ? Sa devyin incrouayab, pazordinèr, ranversan, sa vouzaalor indsé droldaspé dontonrevyinpa... Epui sané ancor ryin, sané ke demimezur... ».

Queneau poursuit en introduisant

pour	*ch*	*gn*	*ou*	*eu*	*an*	*in*	*on*	*un*
la notation	$\breve{c}$	$\tilde{n}$	*w*	*oe*	$\tilde{a}$	$\tilde{i}$	$\tilde{o}$	$\tilde{u}$

Concevoir l'automate transcrivant un texte usuel en texte « à la Queneau », dans ce mode complété. L'opération inverse serait-elle facile ? Se reposer ces questions dans le cadre de la *reconnaissance automatique de la parole.*

16.9.4

Toute interconnexion en cascade ou en parallèle d'automates à états finis est encore un automate à états finis.

16.10 LECTURES

AHO & ULLMAN, Motifs, automates et expressions régulières, chap. 10 in *Concepts fondamentaux de l'informatique*, Dunod, 1993.

AUTEBERT J.M., *Théorie des langages & automates,* Masson, 1994.

BENZAKEN C., Introduction aux langages formels, grammaires et automates, chap. II in *Systèmes formels ; introduction à la logique et à la théorie des langages,* Masson, 1991.

BERSTEL & REUTENAUER, *Les séries rationnelles et leurs langages,* collection ERI, Masson, 1984.

BOOTH T.L, *Sequential Machines and Automata Theory*, Wiley, 1967.

FAURE & LEMAIRE, Monoïdes & automates, chap. VI in *Mathématiques pour l'informaticien,* vol. 1, Gauthier-Villars, coll. Programmation, 1973.

GROSS M. & LENTIN A., *Notions sur les grammaires formelles,* collection Programmation,Gauthier-Villars, 1967

(Contient quelques indications sur les polynômes associatifs de Schützenberger, et leur application à la quantification des ambiguïtés d'une grammaire.)

MARCHAND, *Mathématique discrète, outil pour l'informaticien*, chap. 8 et 9, De Boeck Université, Bruxelles, 1993.

MICLET L., *Méthodes structurelles pour la reconnaissance de formes,* Eyrolles, 1984

PIN J.E., *Variétés de langages formels,* collection ERI, Masson, 1984.

QUENEAU R., Ecrit en 1937, dans *Bâtons, chiffres et lettres,* Collections Idées, *nrf*, Gallimard, 1965.

INDICATIONS SUR LES EXERCICES & PROBLÈMES

CHAPITRE 0 : MÉMENTO DE LOGIQUE

0.3.0

Partant de $A \wedge B$, opérer par ventilation, adjonction et regroupement.

0.3.1

Prendre A comme hypothèse donne $\neg A$ par détachements successifs, ce qui réfute A mais pas B. Donc A reste condition suffisante sans être nécessaire…

0.3.2

Avec $A \rightarrow B$, $B \rightarrow P$ comme germe, prendre $A \vee B$ comme hypothèse locale et construire un dilemme.

Avec P, hypothèse locale, opérer détachements, conjonction, regroupement déductif.

Toute proposition qui implique P en est condition suffisante : si elle est vraie, P s'obtient par détachement ; de même pour leurs adjonctions.

Toute proposition impliquée par P en est condition nécessaire, ainsi que leurs conjonctions.

Il apparaît maintenant une circularité d'implications entre $(A \vee B)$, P, $(X \wedge Y)$, qui par syllogisme introduit l'équivalence de ces énoncés :

$$R \rightarrow S,\ S \rightarrow T,\ T \rightarrow R \Rightarrow R \rightarrow T,\ T \rightarrow R \Leftrightarrow T \leftrightarrow R$$

0.3.4

Gestion de la contradiction dans les systèmes usuels.

Si un germe G et une hypothèse locale H entraînent une contradiction, le germe est présumé cohérent, et la contradiction éliminable en niant l'hypothèse ; $\neg H$ est alors une proposition déduite de G.

Si un germe entraîne une contradiction sans hypothèse locale particulière, c'est qu'il est contradictoire ; soit il modélise un système existant réellement, et cette modélisation est fausse, soit il modélise exactement un système hypothétique, et celui-ci ne peut exister.

Si une contradiction est dérivable *ex nihilo*, c'est le système de règles d'inférences qui est contradictoire /inconsistant (logique invalide du fait par exemple de règles d'inférence secondaires erronées, ou moteur d'inférence bogué).

0.3.5

Développer l'équivalence puis ventiler la conjonction, et détacher. On obtient ainsi une forme secondaire du principe de détachement.

0.3.6

Montrer que

$A \leftrightarrow B,\ B \rightarrow C,\ C \leftrightarrow D,\ D \rightarrow \neg A \Rightarrow A \rightarrow B,\ B \rightarrow C,\ C \rightarrow D,\ D \rightarrow \neg A$ puis

$A \rightarrow B,\ B \rightarrow C,\ C \rightarrow D,\ D \rightarrow \neg A \Rightarrow \neg A$ avec A comme hypothèse locale.

Puis montrer que $A \leftrightarrow B,\ \neg A \Rightarrow \neg B$.

0.3.7

Procéder par éliminations successives. Voir aussi Jean Julo, *Représentation des problèmes et réussite en mathématiques*, coll. Psychologies, Presses Universitaires de Rennes, 1994, et la démonstration Zebra.pro du système Visual Prolog de PDC.

0.3.9

Montrer que $P(n) \rightarrow P(n+1)$ en remarquant que si $a = b$ alors $2a + 1 = 2b + 1$.

$2^{64} = 10^{64 \cdot 0,30103} \approx 10^{19,3}$; 10^{19} grains = 10^{18} grammes = 10^{15} kilos = 10^{12} tonnes = 10^{6} millions de tonnes...

0.3.10

Contraposer c en c1 : *Ceux qui portent du rouge n'élèvent pas de taureaux,* puis en considérant chaque énoncé comme une implication, construire e, f, h, i, g, j par syllogisme ; g, i, j s'obtiennent plus directement comme résultats de sorites tels que *d, b, c1* $\Rightarrow g$ [L. CAROLL].

0.4.1

Par l'absurde : $\sqrt{2}$ étant présumé rationnel, on pose $\sqrt{2} = p/q$ qui entraîne $p^2 = 2q^2$, à rapprocher de la propriété « tout facteur premier d'un carré parfait l'est un nombre pair de fois » (Euclide).

0.4.2

Soit N *le plus grand nombre premier*, et p_i le i-ième nombre premier ; examiner la divisibilité de $P = 2 \cdot 3 \cdot \ldots p_i \cdot \ldots \cdot N$, puis celle de $P + 1$.

0.4.4

Si f est callimorphe, $-f$ l'est, et $-f$ atteint son maximum sur le pourtour du domaine. Or $\min(f) = -\max(-f)$, et f atteint son minimum où $-f$ atteint son maximum.

Ainsi, la valeur de la fonction dans le domaine est encadrée par les 2 valeurs extrêmes qu'elle atteint à la frontière du domaine.

Si ces valeurs sont égales (du fait de la constance de f sur la frontière F) alors f est constante sur le domaine D:

0.4.6

Voir le problème 0.4.4.

En introduisant une constante k = intelligence(CamaradePremierSecrétaire),

$\forall i \in$ Ivanbourg,	$k \leq$ Intelligence(i)
$\forall i \in$ Ivanbourg,	Intelligence(i) $\leq k$
d'où $\forall i \in$ Ivanbourg,	Intelligence(i) $= k$

0.4.7

g	$\equiv$	gagner les élections
r	$\equiv$	réduire le chômage
e	$\equiv$	développer l'enseignement professionnel
f	$\equiv$	développer les fonds publics

t ≡ taxer
p ≡ privatiser

mènent au germe 0.1-0.5, qui entraîne la démonstration 0.6-0.13

0.1	$g \to r$	
0.2	$r \to (e \wedge \neg t)$	
0.3	$e \to f$	
0.4	$f \to (t \vee p)$	
0.5	g	
0.6	r	détachement 0.1, 0.5
0.7	$e \wedge \neg t$	détachement 0.6 0.2
0.8	e	ventilation 0.7
0.9	$\neg t$	ventilation 0.7
0.10	f	détachement 0.8, 0.3
0.11	$t \vee p$	détachement 0.4 0.10
0.12	$\neg t \to p$	élimination adjonction 0.11
0.13	p	détachement 0.9 0.12

Pour les impôts, ils peuvent être réduits, inchangés, augmentés : si on les réduit, on ne les augmente pas ; mais si on ne les réduit pas, ou ils augmentent, ou ils restent inchangés.

CHAPITRE 1 : ENSEMBLES & ÉLÉMENTS

1.11.4

L'ensemble P(X) regroupe les 2^n parties de X. P(P(X)) est formé de 2^{2^n} collections distinctes de ces parties. Toute couverture de X est collection de ses parties, mais la condition de couverture $\cup C = X$ exclut des couvertures de X certaines collections de parties de X. Donc X possède moins de 2^{2^n} couvertures. On peut cependant montrer qu'il en possède plus de 2^{2^n-1}.

1.11.5

Confronter $A \cap C \subseteq A \subseteq A \cup B$ avec $A \cap C = A \cup B$ donne $A \cap C = A = A \cup B$.

1.11.6

$A \cap C = A - (A - C)$; $A' = R - A$; $(A \cup B)' = (R - A) - B$ ...

1.11.7

Europe des 6 : Allemagne, Bénélux, France, Italie. 4 langues officielles : allemand, français, italien, néerlandais.

Union européenne en 2001 :

- 15 pays : Allemagne, Autriche, Belgique, Danemark, Espagne, Finlande, France, Grande-Bretagne, Grèce, Irlande, Italie, Luxembourg, Pays-Bas, Portugal, Suède ;
- 11 langues officielles : allemand, anglais, danois, espagnol, finnois, français, grec, italien, néerlandais, portugais, suédois.

Chaque pays de Visegrad (Pologne, Tchéquie, Hongrie) introduit une langue ; la Suisse n'en introduirait pas, sauf peut-être le romanche ; la Norvège non plus, si elle se rallie au dano-norvégien.

Un pays unilingue P est *critique* (du point de vue des langues à retenir) si son retrait permettait de retirer la langue L correspondante, c'est-à-dire si c'est *le seul pays unilingue* utilisant cette langue, car c'est alors lui qui rend cette langue obligatoire.

1.11.8

Dans la théorie additive RVB des couleurs, on admet que les lumières se combinent selon les règles :

noir = « absence totale »
azur = vert + bleu
lilas = rouge + bleu
jaune = rouge + vert
blanc = rouge + vert + bleu

que représentent le diagramme de Venn :

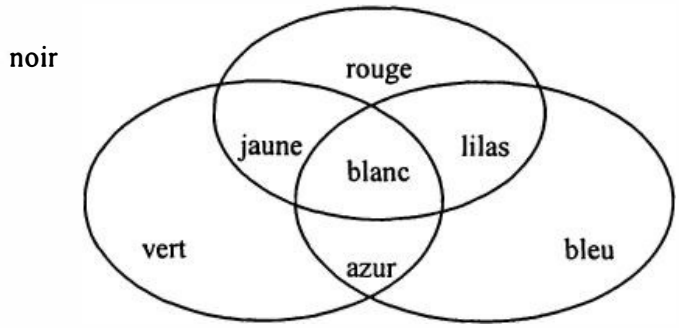


Fig A.1 8 couleurs dérivées de 3 couleurs primaires.

1.12.0

$|A \cup B| = |A| + |B| - |A \cap B|$ donne ici $|A \cap B| = 105 + 145 - 230 = 20$; le trésorier peut donc espérer $210 \cdot 100 + 20 \cdot 150 = 24\,000$ francs.

1.12.1

On a :

$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C| = 230$

$|A| + |B| + |C| = 100 + 70 + 90 = 260$, donc

$|A \cap B| + |A \cap C| + |B \cap C| - |A \cap B \cap C| = 30 = |A \cap B \cap C'| + |A \cap B' \cap C| + |A' \cap B \cap C| + 2 \cdot |A \cap B \cap C|$

Si u est le nombre de membres à inscription unique, d le nombre à inscription double, et t le nombre de membres à inscription triple, alors :

(i) $u + d + t = 230$
(ii) $d + 2t = 30$
(iii) $u, d, t \geq 0$

Les cotisations prévisibles sont alors données par :

$100u + 150(d + t) = 100(u + d + t) + 50(d + t) = 100 \cdot 230 + 50 \cdot (30 - t)$ par (ii).

En notant [a b] l'intervalle de a à b , (ii) et (iii) donnent t = [0 15] et

$100 \cdot 230 + 50 \cdot [15\ 30] = [23.750\ 24.500]$.

1.12.3

$F = C - P$; $G = P - C$; $H = P \cup C$.

$P = H - F$; $C = H - G$.

1.12.5

C'est une *recherche de couverture dégénérée* : aucun expert n'étant unilingue, aucune langue n'est nécessaire.

Gladstone connaissant toutes les langues de Dupont, si une langue permet d'informer Dupont, alors elle permet d'informer Gladstone.

Tout expert connaissant le turc connaît le russe : un tirage en turc n'informerait qu'une partie des personnes informables en russe (2 sur 5).

1.12.6

Une solution élégante ici serait {chinois, français, portugais} (tab. A.1).

Tableau A.1 Construction progressive d'une couverture.

<table>
<tr><td colspan="4">A couvrir : Soyeux, Auvergnats, Mauriciens, Fils du Condor, Kamikazés, Ibères, Goans, Formosans, Luxembourgeois
groupe critique (unilingue) : Auvergnats
langue nécessaire : français
reste à couvrir : Soyeux, Kamikazés, Ibères, Goans, Formosans
groupe critique : aucun ; pivot : Soyeux</td></tr>
<tr><td colspan="2">1er choix : chinois
Reste à couvrir :
Ibères, Goans
critique : aucun
pivot : ibères</td><td colspan="2">2e choix : turc
Reste à couvrir :
Kamikazés, Ibères, Goans, Formosans
critique : aucun
pivot : kamikazés</td></tr>
<tr><td>1er choix : espagnol
Reste à couvrir :
Goans</td><td>2e choix : portugais
Reste à couvrir :
RIEN</td><td>1er choix : chinois
Reste à couvrir :
Ibères, goans</td><td>2e choix : japonais
Reste à couvrir :
Ibères, goans</td></tr>
</table>

Les langues véhiculaires sont à considérer

- si on privilégie un nombre réduit de langues, quitte à ne pas avertir chacun (approche probabiliste),
- si, dans la seconde phase de recherche de couvertures irredondantes, pour trouver parmi les ensembles non redondants restants ceux qui seraient plus utiles que les autres.

1.12.7

Un collège k n'ayant pas de membre en propre est redondant : si on le supprime, tout membre de k garde une raison de rester membre de l'association, puisqu'il appartient à au moins un autre collège.

1.12.8

Reprendre l'exemple du paragraphe 1.9.2.

Traiter couverture et partition comme des listes sans répétition.

La propriété « être disjoint de » étant symétrique, on systématise le passage de la couverture à une partition en rendant progressivement chaque ensemble de la couverture courante, du premier à l'avant-dernier, "disjoint de ceux qui sont à sa droite".

Ce procédé progresse sans remise en cause, c'est donc bien un « algorithme glouton ».

CHAPITRE 2 : RELATIONS BINAIRES

2.6.0

Il existe m·n couples (a, b) $\in$ A⊗B. Pour chacun de ces couples, une relation de $\Re_{AB}$ est soit vraie soit fausse : d'où $2^{m \cdot n}$ relations distinctes de A à B.

De même il existe n·m couples (b, a) $\in$ B⊗A. Pour chacun de ces couples, une relation de $\Re_{BA}$ est soit vraie soit fausse : d'où $2^{n \cdot m} = 2^{m \cdot n}$ relations distinctes de A à B.

On obtient le même résultat en considérant qu'à chaque relation $R \in \Re_{AB}$ correspond une et une seule relation $R^{-1} \in \Re_{BA}$, et réciproquement.

2.6.1

x (R•S) y $\Leftrightarrow$ ($\exists$z xRz $\wedge$ zSy). Pour que (x, y) satisfasse R•S, il faut donc que le point x (de P) appartienne à au moins une droite z (de D), qui soit en contact en un seul point avec le cercle y (de C). Autrement dit, que le point x (de P) appartienne à au moins une droite z (de D), *tangente* au cercle y (de C). Autrement dit, que le point x (de P) :

appartienne à y (de C), et une seule droite z est possible,

ou *soit extérieur à y* : dans ce dernier cas, deux droites z sont possibles.

2.6.2

Ensemble des moteurs employés :

$$N = \{m \mid m \text{ montéSur } v \wedge v \in L\}.$$

Ensemble des modèles utilisant actuellement ces moteurs :

$$L^{+} = \{v \mid m \text{ montéSur } v \wedge m \in N\} \supseteq L.$$

Ensemble des modèles restant achetables à court terme :

$$L^{*} = \{v \mid m \text{ montéSur } v \wedge m \in N - A\}$$

Ensemble des modèles du parc à vendre à court terme :

$$L^{-} = L - L^{*}.$$

L'exemple donné fig. 2.1 / tab. 2.2, et la donnée d'une liste L permettent de trouver successivement les listes M et L^{+}. Puis la liste A restreint L^{+} en L^{*}, d'où l'on déduit L^{-}. (tab. A.2).

Tableau A.2 Gestion d'un parc automobile.

Modèles / Moteurs	141 L	141 N	141 LX	141 S	343 L	343 N	343 LX	343 S	545 L	545 N	545 LX	545 S	N	A
Z1	x													
Z2		x»	x	x»	x»								*	–
Z3						x»	x		x»				*	
Z4								x»		x	x»		*	
Z5												x		–
L »		x		x	x	x		x	x		x			
L^{+}		x	+	x	x	x	+	x	x	+	x			
L^{*}						x	x	x	x	x	x			
L^{-}		–	?	–	–									

2.7.0

On pose <commande> porteSur <objet> / <objet> moulableSur <moule> / <moule> installableSur <machine> .

<commande> exécutableSur <machine> ≡ <commande> (porteSur • moulable-Sur • installableSur) <machine> ≡ ∃ o ∃ m <commande> porteSur o ∧ o moulableSur m ∧ m installable sur <machine>.

On s'en fera une idée en dessinant 3 diagrammes sagittaux « en cascade », l'ensemble d'arrivée du premier (resp. du second) étant l'ensemble de départ du second (resp. du troisième).

2.7.1

On pose <étudiant> utilise <station>, <station> connectée_à <réseau>, <réseau> accède_à <porte>.

Le schéma de communication est du style (utilise • connectée_à • accède_à • débouche_sur • connectée_à$^{-1}$ • utilisée_par).

CHAPITRE 3 : FONCTIONS

3.11.2

Pour une fonction f: $\mathbb{R} \to \mathbb{R}$ paire, toute valeur de im(f) a pour origine soit 0 soit 2 origines de signes opposés, et seule la restriction de f à $\mathbb{R}^+$ pourra être inversible. Ne sont donc inversibles que les fonctions paires de domaine {0}, c'est-à-dire les fonctions f : 0 → k.

3.11.3

Pour k > 0, les fonctions sont totalement définies ; pour k < 0, les fonctions sont définies sur $\mathbb{R} - \{0\}$ mais pas en 0. Elles ne sont inversibles que pour k impair (cf. exercice précédent).

3.11.4

Soit $f : x \to x^2 + 1$;

$f \circ f : x \to y^2 + 1$ où $y = x^2 + 1$ soit $f \circ f : x \to x^4 + 2x^2 + 2$ et

$f \circ f \circ f : x \to z^2 + 1$ où $z = y^2 + 1 = x^4 + 2x^2 + 2$ soit

$f \circ f \circ f : x \to x^8 + 4x^6 + 8x^4 + 8x^2 + 5$.

3.11.5

Le tableau A.3 décrit une fonction n(p, q) simple pour la numérotation systèmatique des points du quadrant $\mathbb{N}\otimes\mathbb{N}$: en effet, la relation n(p, q) = q + n(p + q, 0) nous ramène aux n(p, 0)= p·(p + 1)/2.

Vérifier que cette fonction N constitue une *application injective* $\mathbb{N}\otimes\mathbb{N} \to \mathbb{N}$, d'où le caractère dénombrable de $\mathbb{N}\otimes\mathbb{N}$.

Tableau A.3 Numérotation des sommets d'un quadrillage ouvert en quadrant.

n		q									
		0	1	2	3	4	5	6	7	8	9
	0	0	2	5	9	14	20	27	35	44	54
	1	1	4	8	13	19	26	34	43	53	64
p	2	3	7	12	18	25	33	42	52	63	...
	3	6	11	17	24	32	41	51	62	...	
	4	10	16	23	31	40	50	61	...		

$\mathbb{N}\otimes\mathbb{N}$ peut lui-même être mis en bijection avec l'ensemble des fractions, qui de ce fait est dénombrable, et dont $\mathbb{Q}$ est un sous-ensemble (voir aussi sect. 5.5).

3.11.6

A chacun des k éléments de A on peut associer

a) 1 élément parmi les n de B pour chaque application, d'où n^k applications $A \rightarrow B$, et la relation $|B^A| = |B|^{|A|}$;

b) 1 élément parmi n+1 s'il s'agit d'une simple fonction, le n+1-ème dénotant l'absence d'image, d'où $(n+1)^k$ fonctions de A vers B (dont n^k applications).

3.11.7

Pour borner supérieurement une diagonale, lui appliquer 2 fois l'inégalité du triangle et retenir la valeur la plus contraignante. La borne inférieure s'obtient de même à l'aide de la proposition 3.16.

3.11.8

Vérifier que $\forall x : d_H(x,x) = 0$

$\forall x, y : d_H(x, y) = d_H(y, x)$

$\forall x, y, z : d_H(x, y) \leq d_H(x, z) + d_H(z, y)$

3.11.9

Pour des déplacements x, y *successifs*,

- $d1(x, y) = | x1-y1 | + | x2-y2 |$ est la distance réaliste en durée si on suppose un mouvement uniforme, négligeant le départ et l'arrêt,
- $d1/2'(x, y) = \sqrt{| x1-y1 |} + \sqrt{| x2-y2 |}$ est la distance réaliste en durée pour des déplacements formés d'un mouvement uniformément accéléré puis décéléré.

Pour des déplacements (x, y) *simultanés*, considérer $d_\infty(x,y) = \underset{i=1 \text{ à } 2}{\text{MAX}} | x_i-y_i |$.

Tableau A.4 Estimation de la durée de déplacement d'une tête traçante.

déplacements	successifs accélérés/décélérés	successifs uniformes	simultanés
	$d_{1/2}'$	d_1	d_∞
(8, 8)	5,66	16	*8*
(12, 4)	5,46	16	12
(16, 0)	*4*	16	16
	privilégie les déplacements parallèles aux axes		*privilégie les déplacements diagonaux*

3.11.10

Prendre $d_\infty(x,y) = \underset{i=1 \text{ à } 2}{\text{MAX}} | x_i-y_i |$.

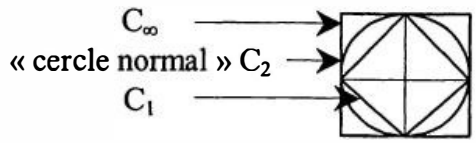


Fig. A.2 Cercles généralisés.

3.12.1

Tout nombre n peut être «deviné» sans ambiguïté si l'on en possède une image f(n), où f est inversible.

Ici, à n entier strictement positif on associe $r = n\cdot(n-1)/2$, toujours entier ; car ou n est pair, ou $n-1$ l'est. On en tire $2r = n\cdot(n-1)$ soit $n^2 - n - 2r = 0$ et $n = ½ + \sqrt{(2r+½)}$, valeur entière en l'absence d'erreur. En fait, le « devin » se contentera probablement d'approcher n par $n = \lceil\sqrt{2r}\rceil$ et de regarder si « ça colle ».

Tableau A.5 Résolution d'une devinette numérique.

Stratégie	théorique			probable		
Réponse r	$2r + ½$	$\sqrt{}$	$n = ½ + \sqrt{}$	$2r \nearrow n^2$	$n = \lceil\sqrt{2r}\rceil$	$r = n\cdot(n-1)/2$
55	110,25	10,5	11	110↗121	11	55 !
300	600,25	24,5	25	600↗625	25	300 !
75	150,25	12,257...	*non entier*	150↗169	13	78 ?

Le problème du devin est de *trouver une fonction injective* suffisamment simple pour que le joueur arrive à calculer r, et suffisamment complexe pour qu'il ne comprenne pas l'opération inverse.

EXEMPLES

« Penser à un nombre entier entre 10 et 90 [$n = 10d + u$] ; lui ajouter le nombre des dizaines [$n' = 11d + u$] » définit une fonction résultat $n' : \mathbb{N} \to \mathbb{N}$ injective ;

- les résultats $11d + 10$ comme 21, 32, 43, ... sont hors d'atteinte sans erreur,
- sinon, $n = n' - d = n' - \lfloor n'/11 \rfloor$; ainsi, $n = 37 \Rightarrow n' = 40 \Rightarrow n = 40 - \lfloor 40/11 \rfloor = 40 - 3 = 37$.

Par contre « Penser à un nombre entier entre 10 et 90 [$n = 10d+u$] ; lui enlever le nombre des dizaines [$n' = 9d + u$] » mène à une ambiguïté des résultats multiples de 9 ; ainsi : $45 = 9\cdot4 + 9$ (origine : 49) $= 9\cdot5 + 0$ (origine : 50).

Que f ne soit pas surjective donne au devin une capacité de détection d'erreur.

3.12.2

Sous certaines hypothèses simplificatrices, la relation peut être fonctionnelle pour les couples (groupe, heure), (enseignant, heure), (salle, heure). Dans chacun de ces cas, l'emploi du temps se ramène à une table rectangulaire de triplets, telle que

$$\text{(groupe, heure)} \to \text{(activité, enseignant, salle)}$$

Du fait des programmes, seule une partie de $|G\otimes A|$ est utile ; du fait des spécialisations des enseignants, il en est de même de $|E\otimes A|$... Avec 60 enseignants et 200 activités, il n'y a peut-être que 300 à 500 combinaisons à envisager parmi 12 000 ! Développer ce genre de remarques pour établir que le nombre d'emplois du temps admissibles est faible devant $|G\otimes H\otimes A\otimes S\otimes E|$.

La validité des emplois du temps suppose aussi qu'il n'y ait pas de collisions, c'est-à-dire deux affectations simultanées distinctes pour une même personne (enseignant ou étudiant) ou une même salle. Enfin, le problème est encore réduit avec des clauses du genre « pas plus de 6 heures par jour pour un même groupe ».

La question se complique avec les activités ne respectant pas le découpage en groupe : enseignements en n-ième langue vivante, enseignements à option, ou les activités telles que *travaux pratiques demandant plusieurs enseignants*.

3.12.3

Montrer que l'encombrement des lexiques d'entrée est en $2m \cdot n$, et celui du lexique général en $m \cdot (n + 1)$.

3.12.5

Une fonction $p(x, y)$ est possible, car les 2 évaluations partielles mènent à une même valeur en (4, 2).

Avec $p(x, y) = a + r \cdot x + s \cdot y$, identifier $p(4, y) = 7y - 3$ et $p(x, 2) = 4x - 5$.

Pour la forme $p(x, y) \equiv f(x) \cdot g(y)$, noter que $p(a, y) = Ay + B$ et $p(x, b) = Cx + D$ ne peuvent découler d'une même fonction que si $p(a, b) = Ab + B = Ca + D$.

Alors, $p(x, y) \equiv f(x) \cdot g(y)$ est satisfaite par f et g linéaires, si $p(a, b) \neq 0$.

Cette identité donne en effet $p(a, y) = f(a) \cdot g(y)$, d'où $g(y) = (Ay + B)/f(a)$, et $p(x, b) = f(x) \cdot g(b) = Cx + D$, d'où $f(x) = (Cx + D)/g(b)$. Donc $p(x, y) = (Ax + B) \cdot (Cx + D) / (f(a) \cdot g(b))$; $p(a, b) = f(a) \cdot g(b)$ donne finalement $p(x, y) = (Ay + B) \cdot (Cx + D) / p(a, b)$, soit ici $p(x, y) = (4x - 5) \cdot (7x - 3)/11$.

La condition $p(a,b) = Ab + B = Ca + D$ nous ramène bien à $p(a, y) = Ay + B$ et $p(x, b) = Cx + D$.

Dans le cas général $p(x, y) = \alpha + \beta \cdot x + \gamma \cdot y + \delta \cdot xy$, les 2 « expériences » associées dégagent deux formes linéaires (soient 4 coefficients) liées par une même valeur en (a, b). Il ne reste donc que 3 coefficients indépendants, et la reconstitution de $p(x, y) = \alpha + \beta \cdot x + \gamma \cdot y + \delta \cdot xy$ est ainsi sous-déterminée. Une troisième coupe serait nécessaire pour une solution sans ambiguïté.

La première solution lève l'incertitude en annulant le coefficient de xy.

La solution construite en $f(x) \cdot g(y)$ fait partie des solutions « à variables séparées » ; elle lève l'incertitude en posant $q \cdot t = r \cdot s$ et n'est qu'une autre des solutions compatibles avec les 2 expériences.

CHAPITRE 4 : RELATIONS BINAIRES INTERNES

4.8.0

Utiliser la propriété que $p \rightarrow q$ est toujours vrai quand p est faux.

4.8.1

$\perp$ est irréflexive, intransitive, symétrique.

4.8.2

Réflexivité, antisymétrie, transitivité dans A^n se déduisent de la propriété dans A.

4.8.3

Revoir l'ordre strict.

4.8.4

$\forall x, y \in E\ x\pi y \equiv d(x, y) \leq r \Rightarrow \forall x, y, z \in E\ (x\pi y \wedge y\pi z) \rightarrow d(x, y) \leq r \wedge d(y, z) \leq r$ d'où, par l'inégalité du triangle $d(x, z) \leq d(x, y) + d(y, z)$:

$$\forall x, y, z \in E\ x\pi y \wedge y\pi z \rightarrow d(x, z) \leq 2r$$

qui ne permet pas de garantir la transitivité, car $x\pi z \equiv d(x, z) \leq r$.

Mais si $d(x, z) \leq \max(d(x, y), d(y, z))$ alors $d(x, z) \leq r$, et la transitivité est garantie. Pour une distance ultramétrique, l'analogie π devient équivalence.

4.8.5

Circularité + symétrie → transitivité ; transitivité + symétrie → circularité ; il est donc équivalent de dire une *équivalence*

- réflexive, symétrique, transitive,
- réflexive, symétrique et *circulaire*...

4.8.6

Cas type de préordre. Permet de gérer les formules trouvées ; la première formule trouvée est gardée ; pour toute autre formule :

- si elle est de défaut strictement supérieur à celui des formules en stock, on l'oublie ;
- si elle est de défaut strictement inférieur à celui des formules en stock, elle les remplace ;
- si elle est de défaut égal à celui des formules en stock, on la stocke également.

On peut aussi contraindre le procédé de recherche à chercher/trouver des formules strictement meilleures que celle(s) connue(s) ; à chaque succès, on révise à la baisse le défaut autorisé dans la suite de la recherche.

4.8.7

Un tri se justifie d'abord par un *ordre total* permettant de toujours décider comment ranger 2 éléments différents. Si les éléments considérés forment une suite dont les répétitions ne sont pas exclues, l'ordre total sur les valeurs possibles se réduit à un préordre total sur les éléments de la suite (équivalents s'ils ont même valeur sans avoir même rang).

Pour les couples (clé, info), ils sont soumis à un *préordre total* dès que l'ensemble des clés est totalement (pré)ordonné ; alors, des couples (clé, info) de même clé sont ex aequo tout en pouvant être distincts ; et on sait dans quel ordre mettre les couples ayant des clés distinctes.

4.8.8

// est une équivalence (préordre symétrique) si on accepte qu'une droite soit parallèle à elle-même ; ses classes d'équivalence sont les *directions* du plan.

$\perp$, étudié à l'exercice 4.8.1, n'appartient à aucune grande classe de relations.

4.8.9

Une tâche ne se précède pas elle–même, d'où l'irréflexivité.

Si t12 précède t4, et t4 précède t37, il est clair que t12 précède t37, d'où la transitivité.

D'autre part, si t12 précède t4, cela interdit que t4 précède t12. Pour un jeu de tâches à ordonnancer, les contrôles de *vraisemblance des données* devront précisément *vérifier qu'il n'y a ni symétries ni circularités* dans les précédences données.

4.8.10

La représentation virgule flottante est souvent définie pour assimiler l'*ordre entre nombres virgule flottante* à un *ordre lexicographique portant sur des triplets (signe, exposant, mantisse)* de valeur $\text{signe} \cdot 2^{\text{exposant}} \cdot \text{mantisse}$

On suppose l'unicité de représentation.

- *Comparer les signes* achève la comparaison s'ils sont distincts.
- S'ils sont égaux, *on compare les exposants* : l'exposant le plus grand correspond au nombre le plus grand en valeur absolue ; en cas d'égalité des signes *puis* des exposants, il reste à *comparer les mantisses pour trancher.*

Si ces 3 tranches sont implantées dans un ordre convenable, 2 nombres virgule flottante ont entre eux la même relation <, =, > que les entiers de même représentation binaire.

Dans les processeurs correspondants, la même instruction binaire compare indifféremment une paire d'entiers ou une paire de nombre virgule flottante, évitant ainsi le recours à une soustraction flottante.

4.8.11

On considère un ensemble P de personnes, muni de la relation « aime » ;

- sous une hypothèse de fraternité universelle, « aime » est transitive,
- sous une hypothèse de jalousie universelle, « aime » est intransitive,
- dans tout tiers cas « aime » est non transitive.

La réflexivité traite alors de « l'amour de soi » (pour chacun, pour aucun...)

- si « aime » est *symétrique*, tout amour est réciproque ;
- si « aime » est *asymétrique*, aucun amour n'est réciproque, allant jusqu'à interdire l'amour de soi ;
- si « aime » est *antisymétrique*, l'amour réciproque se limite à l'amour de soi.

4.8.12

Voir le problème 0.4.6. Le point de départ de Zinoviev (*Les hauteurs béantes,* L'Age d'Homme) est une échelle d'intelligence, et 2 arguments opposés tendent à prouver que *tous les habitants d'Ivanbourg sont équivalents pour cette échelle.*

On ne peut en conclure qu'ils sont identiques, malgré ce point commun (cf. discussion ordre ≠ préordre). Il y a donc probablement un prototype de l'Ivanbourgeois moyen, qu'on peut appeler Ivan Ivanov. *En introduisant de nouveaux critères indépendants du premier, on retrouverait la variété des individus.*

4.8.14

Se ramener aux matrices booléennes carrées représentant les relations, en distinguant les 5 cases de la diagonale, les 10 du triangle sous-diagonal, les 10 du triangle sur-diagonal. Etablir ainsi qu'il y a sur cet ensemble 2^{25} relations distinctes dont 2^{20} réflexives, autant d'irréflexives...

4.9.1

Voir les exercices 4.8.4 et 4.8.13.

A partir de la partition discrète, qui isole chaque élément, on voit apparaître les classes {a, e}, {c,d}, {a, e, c, d} et enfin {a, b, c, d, e} correspondant aux boules de rayons 5, 7, 9, 15.

4.9.2

$\forall x,y \in E\ xP_ky \rightarrow xP_{k-1}y$.

Supposons chaque utilisateur *cohérent*, en ce sens que *s'il préfère x à y et y à z, alors il préfère x à z.*

S'ils sont p utilisateurs à préférer x à y et q à préférer y à z, alors ils sont *au moins* $s = \max(p + q - N, 0)$ qui préfèrent x à z (cf. proposition 1.20).

Pour chaque case P[x, z] , la sous transitivité imposera donc :
$P[x,z] \geq \max_y (P[x,y] + P[y,z] - N)$.

4.9.3

Voir le problème 4.9.2. Avec l'expression $s = \max(p + q - N, 0)$, la question est de savoir, pour une majorité M donnée si $p \geq M$ et $q \geq M$ entraînent

- $s \geq M$ (vote cohérent),
- $N - s \leq M$ (vote non contradictoire).

4.9.4

L'égalité est plus fine que toute congruence, toute congruence modulo n est plus fine que les congruences modulo un facteur de n ; dans cet ordre, les congruences les moins fines sont donc modulo un nombre premier ; au-delà, la congruence modulo 1 est la relation universelle entre entiers, car leur différence sera toujours divisible par un.

Des implications ci-dessus, on déduit aussi que pour 2 congruences modulo k et n, la congruence la moins fine plus fine que chacune des deux sera la congruence modulo ppcm(k,n) et la congruence la plus fine qui soit moins fine que chacune d'elles sera la congruence modulo pgcd(k,n). Ainsi, les congruences modulo 12 et 20 sont-elles encadrées par la congruence modulo 60 (borne inférieure), et la congruence modulo 4 (borne supérieure).

Soit à résoudre $m^2 + n^2 = 36mn + 12m - 18n$.

Toute solution de cette équation est solution de

$$m^2 + n^2 \equiv (36mn + 12m - 18n) \bmod k \quad \forall k \geq 2$$

Choisissons $k = 6$, qui annule le second membre, et nous ramène à la congruence

$$m^2 + n^2 \equiv 0 \bmod 6$$

Cette congruence a pour solution $m \equiv 0 \bmod 3$ et $n \equiv m \bmod 6$: c'est donc parmi les couples ($m = 3q$, $n = 3q + 6p$) qu'il faudra chercher les solutions de l'équation initiale.

4.9.5

$6 = 2\cdot3$, $210 = 2\cdot3\cdot5\cdot7$, $30\,030 = 2\cdot3\cdot5\cdot7\cdot11\cdot13$ mais $25 = 5\cdot5$ et $180 = (2\cdot2)\cdot(3\cdot3)\cdot5$.

Les bornes sont données par *ppcm* et *pgcd*. Tout nombre du treillis issu de Q peut être écrit comme produit des facteurs premiers de Q, ayant chacun une puissance 0 ou 1 selon qu'il est inemployé ou employé. Dès lors qu'on assigne un rang à ces facteurs premiers, il y a bijection entre un *vecteur booléen* et le nombre admettant ce vecteur comme *suite des exposants de ses facteurs premiers.*

Comme $30\,030 = 2\cdot3\cdot5\cdot7\cdot11\cdot13$, il y aura bijection entre les diviseurs de 30 030 factorisés sous la forme standard $2^p\cdot3^q\cdot5^r\cdot7^s\cdot11^t\cdot13^u$ et les vecteurs booléens (p q r s t u), *pgcd et ppcm* étant images arithmétiques de *max/min* appliqués aux vecteurs booléens composante à composante.

$$\begin{aligned}\text{ppcm}(21, 30) &= \text{ppcm}(2^0\cdot3^1\cdot5^0\cdot7^1\cdot11^0\cdot13^0,\ 2^1\cdot3^1\cdot5^1\cdot7^0\cdot11^0\cdot13^0)\\ &= 2^1\cdot3^1\cdot5^1\cdot7^1\cdot11^0\cdot13^0 = 210\\ &\Leftrightarrow \max((0\ 1\ 0\ 1\ 0\ 0), (1\ 1\ 1\ 0\ 0\ 0)) = (1\ 1\ 1\ 1\ 0\ 0).\end{aligned}$$

4.9.6

Cf. LIVERCY, *Théorie des programmes*, Dunod.

4.10.0

Cf. A.THAYSE et coll., *Approche logique de l'intelligence artificielle*, Dunod, tome 2.

Le quantificateur $\exists!$ signifie : il existe un et un seul ; on le développe par des équivalences telles que :

$$\exists!\, x\, P(x) \Leftrightarrow \exists\, x\, P(x) \wedge (\forall y \forall z\, P(y) \wedge P(z) \rightarrow z = y)$$

où le premier prédicat de la conjonction garantit l'existence, et le second l'unicité.

4.10.1

Cf. N.VOROBIEV, *Caractères de divisibilité,* coll. Introduction aux mathématiques, Editions de Moscou, 1973

1/ Utiliser une forme « une propriété est vraie pour tout nombre si

- vraie pour deux nombres, elle est vraie pour leur produit,
- elle est vraie pour ses facteurs premiers ».

2/ Examiner le cas des n^k.

4.10.2

La théorie 1 ne mène nulle part car elle n'expose qu'une propriété *typique* suivie d'un raisonnement cursif qui cache une faille : ... si je divise le nombre de trous par 2... Il est clair que l'on ne sait pas ce qui se passe si le nombre de trous est impair, et, en particulier égal à 1 (cas improbable que la définition n'exclut pas).

La théorie 2 est d'une rédaction au départ classique, dans laquelle « ordonner » se réfère de fait à un préordre. La difficulté levée à la fin n'est pas résolue.

La théorie 3 est lourde mais plus propre.

La question des tamis & filtres peut être traitée en supposant un nombre de trous élevé, ou mieux un ensemble des trous « suffisamment dense » ce qui posera la question d'affiner la définition des passoires, par exemple à l'aide d'une distance intertrous.

Voir aussi http://gabuzomeu57.free.fr/pass.htm.

4.10.3

Le modèle de Ptolémée a été abandonné car moins *pertinent* que celui de Newton : pour expliquer à finesse égale les mouvements de 7 planètes, il faut 7 ellipses chez Newton, 56 orbes circulaires chez Ptolémée, modèle plus lourd à qualité égale.

Le modèle d'Einstein a été (provisoirement) adopté car pour l'anomalie d'observation du périhélie de Mercure, le modèle relativiste explique les 2/3 de l'écart constaté entre observation et prédiction newtonienne : d'autre part, il se confond asymptotiquement avec le modèle de Newton aux basses vitesses (i.e. aux vitesses inférieures au dixième de la vitesse de la lumière).

Les modèles successifs d'un système forment une suite, dont la convergence vers un modèle parfait reste potentielle, la simplicité demandée par l'utilité du modèle contrecarrant l'exigence de précision.

CHAPITRE 5 : FONCTIONS, CALCULABILITÉ, RÉCURRENCE

5.6.0

Seul 2 est pair et premier.

Aucun nombre impair n'a de diviseur pair.

Si n est divisible par un nombre non premier $p = q \cdot \ldots$, alors il fait partie des nombres divisibles par tout facteur q... de ce nombre. Inversement, un nombre non divisible par q fait partie des nombres non divisibles par les multiples p de q. La non-divisibilité d'un nombre n par un nombre non premier $p = q \cdot r$ est donc assurée si la non divisibilité par q (resp. r) l'est. Pour établir la primalité comme conjonction d'indivisibilité par une suite de nombres, il suffit de l'établir pour une liste réduite, dont on a éliminé tout nombre dont on garde les facteurs. Partant de la suite des entiers successifs de 2 à m, on montrera par descente récursive que l'application réitérée de ce procédé ne laisse subsister que les nombres premiers de 2 à m (cf. crible d'Erathosthène).

Soit n le nombre, et p un de ses diviseurs ; alors il existe q tel que $n = q \cdot p$; si $1 < p \leq \sqrt{n}$ alors $\sqrt{n} \leq q < n$, et si $\sqrt{n} \leq p < n$ alors $1 < q \leq \sqrt{n}$; donc, si n admet un diviseur p entre 1 et n, alors n admet un diviseur min(p, n : p) entre 1 (exclus) et $\sqrt{n}$ inclus.

Nb de tests : définition initiale : 956 ; 2) 478 ; 3) 162 ; 4) 31 ; 5) 11 ; 6) 15.

5.6.3

Pgcd selon Euclide. Le cas 0, inconnu à l'époque, n'est pas traité.

5.6.4

Pgcd : définition rapide, incluant le cas 0 (dont tout entier n est diviseur puisque $n \cdot 0 = 0$).

5.6.5

Essayer YiKing ≡ C.

5.6.6

Etudier le déroulement de Legendre (n, 37) = n·Legendre(n, 36) = ...

Montrer par récurrence que Legendre (n, k) $\equiv n^k$.

5.6.7

On code un entier *n* avec *n + 1* bâtonnets consécutifs, car zéro est ainsi représenté par un bâtonnet – ce qui le distingue d'un blanc (ou vide).

On suppose la tête de lecture en T, dans l'état 0 ; on passe tous les blancs jusqu'au premier bâtonnet à droite ; ce premier groupe est formé de m + 1 bâtonnets et le suivant de n + 1 ; en transportant m bâtonnets du premier au second groupe, il reste 1 et m + n + 1 bâtonnets, représentant 0 et m + n : on conclut en effaçant le dernier bâtonnet du premier groupe, et en arrêtant la machine ; le « transport » d'un bâtonnet du premier au second groupe est réalisé en effaçant le dernier du premier groupe, puis en l'écrivant dans la dernière case de la zone blanche, en tête du second groupe.

- La multiplication suppose sur le ruban un premier groupe de m + 1 bâtonnets (multiplicateur), un second formé de n + 1 bâtonnets (multiplicande) et une grande zone blanche au-delà, dans laquelle on installe le produit p en formation en déposant un bâtonnet isolé ; alors, tant que le premier groupe possède plus d'un bâtonnet, il faut enlever un bâtonnet de ce groupe, puis ajouter le second groupe au troisième *sans détruire le second* ; on ajoutera ainsi m fois le second groupe (valant n) au troisième groupe initialement nul, ce qui donnera dans le troisième groupe $m \cdot n + 1$ bâtonnets représentant le produit $p = m \cdot n$ cherché.

- Pour régler le problème de l'*addition non destructive* nécessaire à la multiplication, il faut reprendre l'addition initiale en remplaçant l'effacement d'un bâtonnet par sa transformation en astérisque par exemple, * devenant la trace d'un bâtonnet traité ; l'utilisation dans la multiplication supposerait la transformation $|\rightarrow *$ du multiplicande dans chaque phase de cumul, et une restauration $*\rightarrow|$ entre deux cumuls.

- La multiplication est chère, et l'élévation au carré s'inspirera plutôt de la méthode de Töpler, basée sur la formule $(n + 1)^2 = n^2 + 2n + 1$.

5.7.0

Etudier le déroulement de estPremier(11) et estPremier(49). S'appuyer sur la propriété : tout nombre premier autre que 2 est impair.

5.7.1

Calculer la trace de premier(6) puis premier(9).

CHAPITRE 6 : NOTION DE COMPLEXITÉ

6.8.0

La durée d'évaluation, la mémoire utilisée sont en O(n), car le phénomène central est

- pour $n > 0$, la transition $n \rightarrow n \bmod 5$ en n/5 étapes, finale sauf si $n \equiv 4 \bmod 5$, qui rajoute 5 appels : $4 \rightarrow -1 \rightarrow 16 \rightarrow 11 \rightarrow 6 \rightarrow 1$;
- pour $n < 0$, la transition $n \rightarrow n \bmod 17$ en $|n|/17$ étapes, qui nous ramène au cas précédent avec de n de 0 à 16, concluant en 1 à 7 étapes (pour n = 14).

Il suit que les grands nombres sont plus efficacement traités s'ils sont négatifs, le traitement restant en tous cas en Θ (n).

Quelle que soit la complexité de l'évolution de l'argument, les calculs de f_{91} ne portent jamais sur le résultat, qui est toujours 91. La forme optimale de f est donc $f_{91}(n) = \{91\}$; alors $f = \Theta(1)$. (☺)

6.8.3

Par récurrence, $C1(k, n) = n! / k ! / (n - k)! = C2(k, n)$. On montre ensuite que la complexité de C1(k, n) est en C(k, n), tandis que C2 est de la classe des O(n – k). La forme C2 est donc beaucoup plus efficace, sauf problème de débordement pour le produit $n \cdot C(k - 1, n - 1)$.

6.8.4

La possibilité de mettre une fonction en table est liée à la nature de l'ensemble de départ :

- oui si l'ensemble de départ est fini, et de taille suffisamment petite ;
- si l'ensemble de départ est dénombrable sans être fini, il faut pouvoir bâtir les valeurs cherchées à partir d'un ensemble fini de valeurs initiales et de procédés d'extrapolation : la fonction est ramenée à une suite amorcée par la table ;
- si l'ensemble de départ est *continu mais borné*, on ne peut définir que les fonctions suffisamment dérivables pour être entièrement définies par un nombre fini de valeurs réparties sur le domaine de définition, complété par un ou plusieurs procédés d'*interpolation* ;

- si l'ensemble de départ est *continu non borné*, on suppose qu'on peut se ramener au cas précédent, complété par des procédés d'*extrapolation*, du style $f(ax + b) = h(f(x))$, exploités sous la forme $f(y) = h(f((y - b)/a))$.

Soit à définir exp(x) avec une précision arbitraire ; on tabule des valeurs pour $x \in [0\ 1]$, et l'on aura par exemple :

$$\exp(0{,}17892) = \exp(0{,}1) \cdot \exp(0{,}07) \cdot \exp(0{,}008) \cdot \exp(0{,}0009) \cdot \exp(0{,}00002)$$

$$= \exp[1] \cdot \exp[17] \cdot \exp[28] \cdot \exp[39] \cdot \exp[42]$$

si on convient que $\exp[10 \cdot i + j]$ contient $\exp(j \cdot 10^{-1-i})$ à la précision demandée, ou

$$\exp(0{,}17892) = \exp(0{,}2)/\exp(0{,}02)/\exp(0{,}001)/\exp(0{,}0001) \cdot \exp(0{,}00002)$$

qui se contentera d'une table deux fois plus petite.

A l'extérieur de la plage privilégiée, on s'y ramène par

$\exp(-x) = 1/\exp(x)$ et pour les grands arguments par
$\exp(x) = \exp(x/2^n) \uparrow 2^n$, obtenu en élevant n fois au carré $\exp(x/2^n)$; ainsi :
$\exp(1000) = \exp(1000/1024) \uparrow 2^{10} = ((((((((((2{,}6553\ldots\ ^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)$
$= 1{,}97 \cdot 10^{434}$.

Si table et algorithme sont également possibles,

- la *table* donne une consultation rapide, moyennant le gel préalable d'une zone mémoire à initialiser correctement, indépendamment de son usage réel : surtout intéressant pour une *utilisation intensive* ;
- sous forme d'*algorithme*, une fonction ne coûte rien au départ, mais peut se révéler lente à calculer : l'algorithme est mieux adapté à l'*utilisation occasionnelle*.

CHAPITRE 7 : DES POINTS ET DES FLÈCHES

7.12.1

A) Tout nombre est divisible par lui-même.
B) Si a divise b, et b divise c, a divise c.

7.12.2

A) Une droite n'est jamais perpendiculaire à elle-même.
B) Si a est perpendiculaire à b, et b à c, a ne l'est pas à c.

7.12.3

En général, G sera non réflexif, non symétrique, non transitif, les propriétés de réflexivités, symétrie et transitivité étant fortes et leurs contraires (irréflexivité, asymétrie, intransitivité) aussi. Ce genre de situation ni ... ni ... est d'ailleurs fréquente en sciences humaines, où les particularités détruisent la plupart des régularités entraperçues.

7.14.0 à 7.14.4

À chaque thème correspond une ligne du tableau A.6 ci-après ; préciser le rôle des arcs peut diversifier le marquage et donner lieu à observations (il s'agit de conventions qui dépendront de la direction dans laquelle on raffine le graphe du modèle).

Tableau A. 6 Modélisations par des graphes.

Domaine	Sommets	Arcs	Marquage	Observations
population	personne	enfant de		
théorie	propositions	implications		
jeu	situation	action, mouvement	nature de l'action, joueur	situations initiales ? finales ?
projet	tâches	précédence	délai	
phrase	notions grammaticales	dépendance	suivi de, s'écrit	

CHAPITRE 8 : CHEMINS & CIRCUITS

8.11.0

Les *chemins* figurent des suites d'appels imbriqués. Les *boucles* rendent compte des récursivités directes, les *circuits* des récursivités croisées (taille 2) ou indirectes (taille > 2). Le circuit élémentaire de taille maximale passant par un point p correspond à la récursivité indirecte la plus complexe que puisse déclencher p.

8.11.3

12 = 3 + 4 + 5. Une permutation de 12 objets à 3 orbes de tailles respectives 3, 4, 5, telle que p = (1 2 3)(4 5 6 7)(8 9 10 11 12) satisfait $p^{60} = p$ car 60 = ppcm(3, 4, 5).

8.11.4

Soit C = (P, Q, R) un chemin formé de 3 sous-chemins consécutifs.

Montrer que ces propriétés sont toujours vraies pour Q (par l'absurde), et étendre aux cas (Q, R) et (P, Q).

8.12.1

On rend d'abord le graphe irréflexif (sans boucle). Alors, si C[x, y] détermine l'existence d'un chemin de x à y, C[y, x] détermine l'existence d'un chemin inverse, et la conjonction, l'existence d'un circuit passant par x et y.

Dans cet ordre d'idée, exploiter le vecteur C[x, *] ∧ C[*, x].

8.12.4

1. Une tâche n'a pas à se précéder : aucune tâche ne doit être sur un circuit de précédence, et le graphe formé doit être sans circuit.

2. La fonction ordinale définie au paragraphe 8.11.2 existe : on l'exploite sous l'une de ses formes (voir aussi sect. 10.6, prop.10.10).

3-4. En assimilant la durée d'une tâche à la longueur d'un arc, l'ordonnancement d'un projet et la détermination de sa durée se rattachent au problème de la recherche dans un graphe du chemin critique ou chemin de longueur maximale,

- généralisation du problème de la recherche du chemin de taille maximale,
- ayant une solution bornée puisque le graphe est sans circuit.

Le calcul des *dates_au_plus_tôt* découle d'un procédé direct (procédant du lancement à la fin). Le calcul des *dates_au_plus_tard* applique un procédé inverse, remontant de la fin au début. *On appelle* ***marge*** *la différence date_au_plus_tard –*

date_au_plus_tôt quand elle existe (i.e. hors chemin critique). Ces marges expriment l'élasticité dont on dispose dans l'exécution du projet.

5. L'accélération du projet se fait par l'accélération des tâches critiques, pour autant qu'elles le restent[1]. Les économies se font sur les tâches présentant une marge, tant qu'elles en ont une. L'application de ces procédés doit être incrémentale, pour faire face à d'éventuels déplacements du(des) chemin(s) critique(s).

8.12.6

Sherman K. STEIN, *Mathematics, the Man-Made Universe,* chap. V : Memory wheels, W.H. Freeman & Co Ed., 1969.

8.12.9

a) Une boucle reflète la tautologie $A \rightarrow A$.

b) $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$ a pour reflet $aUb \wedge bUc \Rightarrow aUc$; le sorite reflète le théorème « dans un graphe transitif, à tout chemin de x à y correspond un arc de x à y ».

8.12.11

L'existence du chemin hamiltonien se montre par récurrence.

Se concentrer sur des tournois à 20 ou 50 compétiteurs plutôt que 3 ou 4.

8.13.0

Voir thèse MACOWICZ.

8.13.2

Chaque personne a en principe un père et une mère, et donc 2^k ancêtres il y a k générations, moins selon le nombre d'unions plus ou moins consanguines.

D'autre part, une population stable correspond en moyenne à 2 enfants par couple.

De Charlemagne à nos jours, chaque couple a eu *en moyenne* plus d'enfants, puisqu'il y a eu expansion démographique *malgré* les guerres et les épidémies. Un de nos ancêtres a donc généralement *plus de* 2^k descendants à la k-ième génération. Sur un même nombre de générations, le « cône des descendants » est donc beaucoup plus peuplé que celui des ancêtres.

CHAPITRE 9 : FERMETURES TRANSITIVES

9.6.0

Suite de l'exercice 8.11.0.

Par opposition à la fermeture réflexo-transitive, la fermeture transitive de G permet de savoir si une procédure p est récursive, indépendamment de la taille du circuit de récursivité, et donc ici comment la compiler.

En passant de la fermeture réflexo-transitive de G à l'équivalence associée puis à son graphe-quotient, on répartit les procédures en classes, regroupables en niveaux. En posant qu'une procédure qui n'en appelle aucune autre est de niveau 0, on peut dire que toute procédure p n'appelle plus que des procédures de sa classe, ou d'une classe de niveau inférieur s'il y en a. Ces classes correspondent à la notion de

---

[1] *En agissant prioritairement sur celles qui seraient communes à plusieurs chemins critiques.*

modules : la récursivité des procédures est possible en chacune d'elles, mais entre elles ces classes forment une hiérarchie.

Ces concepts peuvent être employés pour (re)structurer une application.

9.6.1

Cicuit négatif : prix / volume / embauche / salaires /prix (1 liaison négative).

Circuit positif : prix / salaires / prix. Ce circuit modélise une évolution corrélée des prix et des salaires (→ inflation si hausse, déflation sinon).

Le circuit négatif indique que si un marché est sujet à l'inflation, prix et salaires croissant, les volumes vendus tendent à décroître (du fait notamment de personnes à revenu fixe), freinant les embauches et la hausse des salaires, d'où ralentissement de la hausse des prix, voire baisse des prix et salaires... ce qui pousserait les volumes vendus à croître (du fait notamment de personnes à revenu fixe), accélérant embauches et hausse des salaires, d'où hausse des prix... d'où un cycle économique.

9.6.3

Les fermetures transitives du graphe d'implication

- en tant que processus, correspondent à une exploitation syllogistique des propositions,
- en tant que résultats, nous indiquent si q est déductible de p.

Sont sur un même circuit (dans une même classe) les propositions logiquement équivalentes ; les axiomes sont à chercher dans les classes initiales ; les autres classes regroupent des conséquences de plus en plus lointaines.

Dans le graphe donné, il y a une classe initiale {a, e}, une classe intermédiaire {b, d, f}, et deux classes terminales {c} et {g}.

Toutes ces propositions sont équivalentes si c impliquant a ou e, g implique a ou e ou b ou d ou f ou c (les rôles de c et g étant permutables). Au sens du chapitre 10, il s'agira de rendre le graphe d'implication fortement connexe.

9.6.4

Si les pixels noirs et blancs surgissent de façon aléatoire, chacun étant indépendant du précédent, la longueur moyenne d'une séquence blanche est :

$$1 + 0{,}85\cdot(1 + 0{,}85\cdot(1 + 0{,}85\cdot(1 + \ldots) = 1/(1 - 0{,}85) = 6{,}666\ldots$$

L'observation de séquences d'une longueur moyenne de 20 pixels infirme cette hypothèse. Un modèle markovien à deux états (suite blanche / suite noire) peut expliquer cette longueur. Soit p la probabilité de transition blanc → noir, et 1 – p la probabilité de maintien dans l'état « suite blanche ». La longueur de la séquence blanche moyenne est alors :

$$1 + (1 - p)\cdot(1 + (1 - p)\cdot(1 + (1 - p)\cdot(1 + \ldots) = 1/p = 20\ ;$$

soit p = 0,05 et 1 – p = 0,95. La relation p/q = x/y donne q = (85/15)·0,05 = 0,283... et une longueur moyenne des séquences noires de 1/0,283... = 3,53 pixels.

Supposons maintenant les séquences codées à l'aide de quadruplets : 1 bit pour la couleur, 3 bits pour la longueur (code 000 à 111 mis pour 1 à 8).

La probabilité pour qu'une séquence noire comporte plus de 8 pixels noirs est $0{,}717^8 = 0{,}070$, et une séquence noire sera représentée ici par 4,3 bits en moyenne.

La probabilité pour qu'une séquence blanche comporte plus de 8 pixels blancs est $0{,}95^8 = 0{,}663$, et une séquence blanche sera représentée ici par 11,9 bits en moyenne.

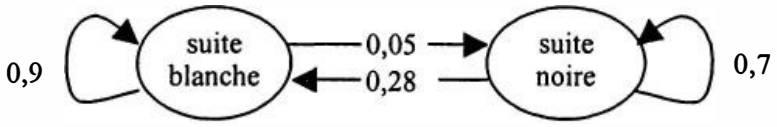


Fig. A.2 Modèle markovien associé à une image noir et blanc, 85% de pixels blancs ; séquences blanches de 20 pixels en moyenne.

Une paire alternée de 20 + 3,53 = 23,53 bits en moyenne est ainsi codée en moyenne sur 11,9 + 4,3 = 16,2 bits, soit une compression de 31%. Une compression un peu plus importante peut être obtenue au-delà de 4 bits, le gain sur les séquences blanches ayant plus d'impact si on garde un format 4 bits (ou même 3) pour les séquences noires.

CHAPITRE 10 : ARBRES & CONNEXITÉ

10.13.1

Il y a une composante fortement connexe, {Bordeaux, Paris, Lyon, Marseille} de centres {Paris, Lyon}. Grenoble et Nice sont périphériques : ils peuvent être informés, mais ne peuvent informer.

10.13.2

Soient x et y les extrémités d'un diamètre, et c un centre. L'inégalité du triangle $d(x, y) \leq d(x, c) + d(c, y)$ se transcrit alors $\delta \leq 2\rho$.

10.13.3

Le graphe d'appel des procédures comporte des boucles ou des circuits s'il y a des appels récursifs. En revanche, la plupart des langages modulaires imposent que les modules forment une hiérarchie : les circuits de récursivité doivent alors rester internes à un module (problème 9.6.0).

CHAPITRE 11 : GRAPHES MULTIPARTIS

11.7.0

Montrer que la suite {2, 5, 11, 23 ...} des $3 \cdot 2^k - 1$ forme le noyau du jeu.

11.7.1

Observer que les nombres premiers sont congrus à 1, 2 ou 3 modulo 4. Revoir le paragraphe 11.3.1 avec k = 3. Montrer par récurrence que la suite des multiples de 4 forme aussi le noyau du nouveau jeu.

La stratégie générale sera donc de toujours se ramener à un multiple de 4 (qui pourra être 0, définitivement gagnant), *qui ne sera jamais premier.*

Si vous recevez

- 4k, vous avez virtuellement perdu : jouez 1, 2 ou 3, et attendez de voir si le vent tourne ;

- 4k + 1, jouez un nombre premier congru à 1 modulo 4 : le nombre lui-même s'il est premier, et vous gagnez immédiatement ; sinon, jouez 1 ou 5 ou 13 ou 17 ou 29...
- 4k + 2, jouez 2, le seul premier congru à 2 modulo 4,
- 4k + 3, jouez un nombre premier congru à 3 modulo 4 : le nombre lui-même s'il est premier, et vous gagnez immédiatement ; sinon, jouez 3 ou 7 ou 11 ou 19 ou 23 ou ...

EXEMPLES

- Vous recevez 15, vous ne pouvez conclure, vous jouez 3 (→ 12), ou 7 (→ 8) ou 11 (→ 4) ; vous vérifierez facilement que, sauf erreur matérielle, dans ces 3 cas vous avez déjà gagné !
- de 10 à 100, appliquez la stratégie générale (objectif : rendre un multiple de 4),
- au-delà de 100, comme 100 est multiple de 4, vous pouvez concentrer votre jeu sur les 2 chiffres de plus faible poids (pour 14 537, on peut jouer comme pour 37) ; vous pouvez aussi utiliser comme accélérateurs quelques grands nombres premiers faciles à retenir : 101 et 103, 601 et 607, 991 et 1009, 10 007...

11.7.3

Se ramener à un problème de coloration pour R^-. On obtient ainsi des clans peut-être amorphes mais sans tension interne. R^+ intervient dans un second temps, comme garantie de solidité. Les cliques ont été expressément définies par BERGE en référence à la sociologie des organisations.

11.7.4

3 couleurs suffisent.

11.7.6

Pour les sommets, 2 couleurs suffisent. Pour les autres questions, s'intéresser au nombre chromatique a) du graphe unissant les milieux des 12 arêtes b) du graphe unissant les centres des 6 faces.

11.7.7

Revoir l'exercice 11.7.4.

Si complexe que soit la maquette, plane par définition, 4 « couleurs » suffiraient à différencier les articles. L'équivalent ici est de 4 polices si on ne joue pas sur les attributs, et avec l'alternance romain/italique, 2 polices seulement.

11.8.2

Le résultat dépend de l'histoire de la partie : le graphe fourni doit être considéré comme le « plateau du jeu », et il faut lui associer un autre graphe modélisant l'histoire ou la dynamique du jeu. Un graphe dont les sommets a, ab, abc, abe, abcd, abce correspondraient aux mouvements faits, serait un arbre infini du fait des circuits du graphe initial. On peut se limiter à la trace utile des mouvements, et raisonner sur un graphe

- dont la position de départ est notée a00,
- qui associe à chaque sommet s du graphe initial des sommets <s, i, j> avec i, j ∈ {0,1} où 0 (resp. 1) indique un nombre pair (resp. impair) d'arrivées en e pour le joueur correspondant.

11.8.4

Les personnes qui se connaissent sont susceptibles de se montrer vos cartes.

Bâtir le graphe de la relation *connaît,* et considérer que son symétrisé G représente la relation *ne doit pas recevoir la même carte que.*

Si la couleur c d'un sommet x représente le type c de carte postale à envoyer à x, dire que le nombre chromatique de G est au plus égal au nombre de types de carte disponibles, c'est dire qu'il existe une coloration résolvant ce problème des cartes postales…

CHAPITRE 12 : OPÉRATEURS ET ALGÈBRES

12.11.9

$X - X = \varnothing$; $X \cap Y = X - (X - Y)$.

12.12.1

Revoir le chapitre 6.

12.12.4

$0 \cdot 1 = 0 \cdot 2$ ne permet pas d'écrire $1 = 2$: la simplifiabilité de $\cdot$ est limitée à $\mathbb{R} - \{0\}$.

12.12.5

Montrer que non : $<\mathbb{B},+> \rightarrow <\mathbb{B}, \cdot>$ est un automorphisme (voir aussi chap. 15).

12.12.6

Distinguer le niveau des opérateurs booléens, les comparaisons, les opérateurs des expressions comparables.

12.12.7

L'idée est qu'un programme est une expression d'une *algèbre des programmes* à base d'instructions, qui renvoie à l'algèbre des expressions pour les arguments des procédures et les affectations. La représentation arborescente des programmes suppose d'expliciter les séquencements figurés par le « ; » séparateur des instructions dans les langages algoloïdes, séquencement implicite en Fortran.

CHAPITRE 13 : MONOÏDES

13.7.10

Pour //, 0 est absorbant, ∞ est neutre, et tout R a (– R) comme symétrique.

De plus, R//R = R/2, R//R//R = R/3 …, d'où P = 3R/2 et Q = (R/3)//(–3R/2) = 3R/7.

Physiquement, il pourra s'agir de calculs sur des résistances : 0 dénote un court-circuit, ∞ l'absence de conducteur, les valeurs négatives correspondent à des dipôles amplificateurs (diodes tunnels…).

13.7.12

Les solutions s'obtiennent en prémultipliant (resp. postmultipliant) membre à membre et en utilisant de même la simplifiabilité à gauche / à droite.

Ex : $axa = b \rightarrow axa \cdot a^{-1} = b \cdot a^{-1} \rightarrow axe = ba^{-1} \rightarrow ax = ba^{-1} \rightarrow a^{-1}ax = a^{-1}ba^{-1} \rightarrow ex = a^{-1}ba^{-1} \rightarrow x = a^{-1}ba^{-1}$.

13.8.0

Cf. DELEDICQ.

1. La transformation identique ;
2. D est un sous-ensemble stable de I ;
3. la composition de 2 anti-déplacements est un déplacement ;
4. i/ une symétrie par rapport à une droite est sa propre inverse,
   ii/ soit 2 symétries s1 et s2, relatives à 2 droites d1 et d2 ; selon que d1 et d2 sont sécantes ou parallèle, la composition de s1 et s2 donne une rotation (y compris l'éventuelle symétrie par rapport au point de concours) ou une translation ; tout déplacement est produit d'une rotation et d'une translation, tout anti-déplacement, d'une symétrie par rapport à une droite et d'une translation...

13.8.2

Considérer chaque dimension M, L, T... comme génératrice d'un groupe monogène d'élément neutre 1, et de là bâtir le groupe des signatures dimensionnelles comme produit cartésien de ces groupes, en supposant tous les produits commutatifs. Un groupe de signatures dimensionnelles à n dimensions est alors isomorphe à $<\mathbb{Z}^n, 0^n, +>$.

Pour les opérations, en supposant S et S′ signatures libres, on distinguera :

- les opérations multiplicatives $\cdot$: $(S,S') \to S{\cdot}S'$, à signatures non contraintes,
- entre grandeurs de même signature, les comparaisons *comp* : $(S, S) \to \mathbb{B}$, et les opérateurs +, –, max, min ... de profil $(S, S) \to S$.

La cohérence de l'utilisation d'une procédure générique se vérifie en contrôlant si les signatures dimensionnelles effectives à l'appel satisfont la contrainte générale de cohérence entre sortes posée lors de la définition de la procédure générique : une procédure effectuant le produit d'une matrice de s par un vecteur de s′ pour donner un vecteur de s·s′ est utilisable pour un produit de matrices de résistances par un vecteur de courants, produisant exclusivement un vecteur de résistances·courants = tensions.

13.9.0

Relire la section 5.5. ou REINHARD.

13.9.2

Abondante production en allemand, notamment à Fribourg ; voir aussi :

MOORE R.E., *Interval Analysis,* Prentice Hall, Englewood Cliffs (USA-NJ), 1966.

NEUMAIER A., *Interval Methods for Systems of Equations,* coll. Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge (UK), 1990.

13.9.3

Ce thème est en relation avec :

- le système Renard, système de valeurs industrielles normalisées, qui exploite l'idée que $\log_r(r^k) = k$; dans ces conditions, les valeurs d'une même série Renard forment un groupe multiplicatif régi par $s_k{\cdot}s_n = s_{k+n}$, et au plan pratique le rang dans la table d'une valeur v, puissance de r, tient lieu de logarithme (entier, à base r) : trouver un produit se ramène à trouver son rang en faisant la somme des rangs des opérandes ; les valeurs de r choisies, racines 5e, 10e, 20e ou 40e de 10, fournissent respectivement 5, 10, 20, 40 valeurs de référence par décade (séries R5, R10, R20, R40) ;

- les séries E (E6, E12...) de valeurs utilisées en électricité, basées sur la notion d'intervalle de tolérance (à 20%, 10%...) ;
- peut-être aussi avec la loi de Weber sur la perception, qui note que deux valeurs physiques x et y ne sont perçues distinctes/équivalentes que si elles diffèrent de 5% au moins/au plus.

Le casse-tête, c'est l'image de l'addition *côté tailles.* En général, $x + y = x\cdot(1 + y/x)$, et on peut traiter l'addition côté taille comme la *correction* de la plus grande des tailles. Le correctif (le nombre de tailles à ajouter à la taille de x), fonction de l'altération y/x, représentée par le nombre de tailles entre x et y, est forcément grossier, et d'autant plus que la raison est grande.

L'associativité ne peut être garantie. Si <+> désigne l'addition calculée par les tailles, il faut s'attendre à ce que l'évaluation de (1<+>1/q)<+>1/q donne un résultat plus faible que 1<+>(1/q<+>1/q), si 1/q est <+>-négligeable devant 1 sans que 2/q le soit.

CHAPITRE 14 : DIOÏDES

14.12.1

Hélas, on n'a pas en général (a // b) + c = (a + c) // (b + c). Donc <ℝ, ∞, 0, //,+> n'est pas dioïde !

14.12.4

D = <$\mathbb{R}^+$, +, min, 0, ∞>.

h ne peut pas recevoir plus de 44 m^3/s, du fait de la saturation de dh (à 26) et eg (à 18).

14.13.2

Supposer transitif le graphe topologique (i.e. dès que le chemin (x,y,z) existe, une corde (x, z) existe).

14.14.1

Une théorie des relations généralisée renvoie-t-elle nécessairement à des ensembles de vérité totalement ordonnés ou tout au moins à structure de treillis ? Les dioïdes permettraient-ils d'exploiter d'autres « systèmes de valeurs » qualifiant des t-uples ? Une théorie des relations généralisée doit-elle renvoyer à une *logique non standard* bien établie, ou peut-elle se contenter de cadres plus faibles tout en restant productive ?

S'attacher à la question de savoir si les relations sont combinables à l'aide d'un dioïde, de préférence idempotent ; considérer par exemple

- la logique de Lukasiewicz L_n, qui suppose une échelle de vérité à n valeurs distinctes,
- la logique floue de Zadeh, avec [0 1] comme échelle de vérité,
- les logiques floues de Chang,
- les logiques floues au sens de Goguen, ayant un *treillis* comme ensemble de vérité,
- de là, d'éventuelles logiques multidimensionnelles, ayant un n-uple pour valeur de vérité (ex : un 4-uple aux champs tels que : degrés de vérité esthétique, économique, politique, morale).

14.14.3

Voir aussi thèse MACOWICZ.

CHAPITRE 15 : ALGÈBRE DE BOOLE

15.12.1

Partir de $x + y = (x' \cdot y')' = 1 \oplus x' \cdot y' = 1 \oplus (1 \oplus x) \cdot (1 \oplus y) \ldots$

15.13.1

Essayer diverses connexions de a, b, c, d, e à 0, 1, x, y ; ex : $f(0, x, 0, y, 0) = x \cdot y$.

15.13.2

Les niveaux successifs du schéma sont $(a, b) \rightarrow (a, a \oplus b, b) \rightarrow (a \oplus (a \oplus b), (a \oplus b) \oplus b) \rightarrow (b, a)$.

15.13.4

Dès qu'on ajoute 1 + 1 bit on a un report ; en général additionner (a) les deux bits donnés pour le poids considéré (b) le report de l'étage (de poids) précédent.

Une multiplication à précision multiple utilise répétitivement le schéma de multiplication complète $mul(A, B, C, D) = A \cdot B + C + D$ où A et B sont des chiffres donnés, C une somme issue d'autres produits partiels de même poids, D un report de l'étage de poids inférieur.

15.13.5

L'additionneur est formé de deux fonctions symétriques : l'ordre des variables n'a pas d'importance. Le multiplicateur est symétrique en x et y d'une part, en a et b d'autre part : son graphe est similaire à celui de l'additionneur si les variables sont groupées en (x y) et (a b), et s'en éloigne sinon.

CHAPITRE 16 : ALGÈBRES DE KLEENE ET AUTOMATES À ÉTATS FINIS

16.7.1

Définir un chemin de transition marqué a, b, b, c, c, c, puis ajouter à chaque sommet les arcs correspondant aux autres cas de figure.

Ou procéder par dérivation de $(a + b + c)^* \cdot ab^2c^3$.

16.7.2

En affectant une entrée a aux minutes avec appel, et s aux minutes sans, il s'agit de définir l'automate associé à (aaa + aasa + asaa + aassa + asasa + assaa) puis à (aa + asa + saa).

16.7.3

Pour casser le code, la première opération sera probablement d'étudier en quoi la distribution des chiffres s'écarte d'une distribution uniforme ; observera-t-on ici une proportion convenable de chiffres, d'abord entre les 0-5 et les 6-9 ? Pour simplifier, raisonner sur un message type correspondant à 1000 caractères.

16.7.5

Amélioration des performances : une analyse markovienne de la langue des textes à transmettre permettrait de minimiser les changements de ligne de la table, qui imposent une insertion du préfixe de la nouvelle ligne.

16.8.0

Utiliser un modèle de Moore, sur la base action = sortie, puisqu'en réalité une action reviendra à

- l'émission de signaux vers un réseau combinatoire ou un automate complémentaire auquel les détails de la phase sont délégués,
- l'attente d'un signal de fin.

On pourra codifier les actions à l'aide d'affectations, M[14] désignant le contenu de la mémoire 14.

16.8.2

Traiter par exemple une expression régulière X = Ee + Ff + Gg, où e, f, g sont des signaux de sortie valant diagnostic.

16.8.3

Ne pas oublier de définir les autres entrées propres au système (porte de cabine ouverte/fermée, franchissements).

16.8.4

Dans ces deux codes, deux combinaisons adjacentes ne diffèrent que par un bit. En frontière de secteur, il peut y avoir une imprécision, mais il ne peut donc pas y avoir d'aléa, combinaison « intermédiaire » à signification erronée introduite par la désynchronisation du basculement de certains bits : de 0111 à 1000 réputés adjacents, 4 bits devraient basculer simultanément ; si un bit bascule d'abord, on obtient à titre transitoire 1111 ou 0011 ou 0101 ou 0110 puis le basculement d'un second introduirait 1011, 1101, 1110... ces combinaisons pouvant dévier au vol le fonctionnement prévu, et fausser le diagnostic voire l'alarme.

16.9.1

Le code Baudot comprend 30 + 2 codes : 2 codes de contexte lettre/chiffre, et 30 codes simples, ce qui permet de fait 60 combinaisons ; avec 5 bits, supposons p codes de contexte et (32 – p) autres codes : on peut coder ainsi $p \cdot (32 - p)$ combinaisons, dont le maximum est 256 pour p = 16.

16.9.2

Un principe similaire peut être appliqué en *reconnaissance automatique de la parole*, où la reconnaissance des phonèmes est souvent ambiguë pour le phonème isolé, ambiguïté levée progressivement sur des séquences, l'identification ayant le plus souvent 1à 3 phonèmes de retard sur l'acquisition.

16.9.3

Suite de l'exercice 16.7.4. Pour une nasale, diverses écritures actuelles mènent au même symbole chez Queneau. Le cas inverse suppose d'autres indications (contexte, étymologie) qui risquent d'excéder les moyens d'un automate à états finis classiques (pour le problème similaire rencontré en reconnaissance automatique de la parole, on a préconisé des automates flous exploitant l'impossibilité de certaines successions de phonèmes dans une langue donnée pour lever les ambiguïtés locales).

16.9.4

Permet de réaliser les automates complexes par composition d'automates simples (ex : le séquenceur de processeur du problème 16.8.0 pourra déléguer à un automate spécialisé l'exécution détaillée d'une phase).

INDEX

#

∀ 12-15
∃ 12-15
∃! 92, 239
( ) 34, 58, 225, 227, 228, 231-236
[ ] 41, 42, 67, 131, 134-136, 145-151
{ } 21-35
*(dualité) 215, 216, 300-304
*(expansion)..... 140, 171, 251, 323
// 86, 240, 257
: 105
= O..... 112, **115**
= o.....115
= Θ.....115
Γ 44, 129, 140
Δ 31
γ 207-209
δ 186
ε 247-255, 263-267, 322-325, 329, 331
ϕ 298, 299
ν 193-196
ρ 186
ω 103, 255, 263-267, 322-325, 329
◭ 6, 9-11, 14-15, 70, 86
′ 28, 133, 313
↔.....3, 5, 55, 238, 313
←.....121
↑ 231
→..... 3-15, 53, 99-103, 237, 254, 255, 313
↓ 236
• 46, 47
o 57
≡ 21
~ 324
≈ 72, 83
| 4, 22, 72
| | 21
ℜ 44, 46-47
℘ 25, 44, 214
⊗ 35, 59
⊕ (somme) 206, 209, 249, 286, 309-313
⊕ (expansion)..... 171, 252, 323
∅ 22, 313
∩ 27, 313
∪ 26, 313
⊄ 23
⊂ 24
⊆ 23, 46, 71, 313
∈ 21
∉ 21
¬ 3, 6-15, 302, 303, 313
∧ 3-15, 302, 303, 313
∨ 3-15, 302, 303, 313
⇔.....5-14
⇒..... 4-14, 254, 255
⌈⌉ 52
⌊⌋ 52
■ .....fin de preuve

A

absorption 236, 304
abstraction 8, 22, 254
absurdité 6, 11, 14, 17
accepteur..... 321, 331
accessibilité 176
additionneur binaire 314
  complet 314
adjonction 3-14, 17, 339, 341
affinogrammes..... 277
aide à la décision..... 80, 81
algèbre
  de Boole.....282-313
  de Kleene 322-325
  des chaînes..... 322
  des relations 44-47, 67-85, 171-178
  des types 260
algèbre formelle 225
  interprétation..... 226
algèbres de termes..... 226
algorithme
  de Dijkstra-Moore 156
  de Kruskal..... 189
  de Legendre **111**, 112, 145, 146, 149, 269, 270
algorithme-glouton..... 40, 155
alliances de nécessité 277
alphabet..... 254, 322
analogie..... 85, 87
analyse dimensionnelle..... 259
analyseur..... 321
ancêtre..... 140
anneau..... **264**, 268
  de Boole..... 264, **310**
annexion d'indéfinis..... 299
antécédents 4, 13

antisymétrie 69
antitonie 75, 283
appartenance à un ensemble 21
application 53, 56
  bijective 55, 75
  de A vers B 53
  de marquage des arcs 135, 265
  multivoque 53, 129
  surjective 75
arborescence 118, **187**, 197, 198
arbre 92, 118, 128, 183, **187**, 189-191, 197, 198
  binaire de décision 306
  couvrant 188
    (de coût) minimal 189, 190
  arc(s) 68, 129,139
arête 69, **134**
arité
  d'un opérateur 223
  d'un prédicat 13
  d'une fonction 13, 61, 225, 237
  d'une relation 60
  fixe 232, 233
  variable 231, 233
arithmétique rationnelle 106
associativité **227**, 231
  à droite 231
  à gauche 231
asymétrie 69
automate
  (multi)graphe de transition 320, 329-331
  à états finis **317**, 321, 328, 333
  à nombre fini d'états *Voir* automate à états finis
  accepteur 321, 331
  analyseur 321
  asynchrone 319
  classifieur 321
  générateur 321, 331
  minimal 176
  non déterministe 331
  réalisation logicielle 320
  réalisation matérielle 319
  synchrone 319
  traducteur 321, 331
automorphisme 238, 283
axiome 226

B

$\mathbb{B}$ *Voir* ensemble des booléens
base clausale 301, 302
  irredondante 303
  minimale 303
  première 302, 303
base de connaissances 300
  d'ordre 0 ..... 302, 308, 312
base de cycles 193
base première 290
  complète 290
  de coût minimale *Voir* minimale
  irredondante 290, 293
  minimale 290
bases de données relationnelles 60, 216
bijection **55**-58, 75
boucle 68, **129**, 141, 319

C

$\mathbb{C}$ *Voir* ensemble des complexes
calcul aveugle 271
calcul en intervalles 90, 260
calculabilité 95
caractérisation d'un opérateur 228
cardinalité 21, 55, 58, 59, 63
carré latin 248
catégorie **226**, 227, 312
ceinture du cube 287
centre d'un graphe 186
cercles généralisés 64
CG(n) *Voir* corps de Galois
chaîne 183, 189, 191
chaîne de finessse entre partitions 34
champ sémantique 272
changement de membre 312
chemin 139, 176, 183
  critique 151, 166
  de coût minimal 151, 152
  de longueur maximale 151, 152
  de longueur minimale 150, 155-157
  de taille minimale 150, 153
  de taille n 139, 145
  élémentaire 141-143, 148
  eulérien 162
  hamiltonien 161
  le moins cher *Voir* de coût minimal
  le plus court *Voir* de longueur minimale
  le plus simple *Voir* de taille minimale
  sans circuit 141
  sous-critique 151
choix multicritère 80-83
circuit 141, 172
  de taille n 141
  élémentaire 142
  eulérien 162
  hamiltonien 161, 162
circularité
  de définition 100
  d'une relation binaire interne 86
classe 227
  de compatibilité 201

de complexité 115
de connexité 184, 185
d'équivalence 74
d'états 174-177
classification
arborescente 34
par analogies et différences 88, 217
classifieur 321, 335
clause **300**, 302, 303
canonique 300, 301
première 302
première inutile 302
première obligatoire 302
première redondante 302
première utile 302
résolvante 301
clé de parité 311
clique 72, 73, **131**, 208, 215, 360
code
DCB 297
de Freeman 258, 277
de Gray 167, 289
détecteur ou correcteur d'erreurs 63, 265
Johnson 297
NaN 53, 103
pondéré 297
saturé 54
commandabilité 176
commutateur 251
commutativité **227**, 240, 251
compilation 256
complément
d'un ensemble 28, 29, 313
d'une relation 133
d'un graphe 133
complexité 112-117
d'un calcul matriciel 124
d'un tri 124
d'une recherche en profondeur 159
d'une recherche par niveaux 154
d'une représentation 123
comportement 317, 321-328
comportement asymptotique 112, 116
comportement fin 174
composante
connexe 184, 185
fortement connexe 185, 196
composition
de fonctions 57
de groupes 250
de relations **46**, 47, 57, 146, 147, 171, 247
composition récursive 100
concaténation **251**, 252, 264
concaténé de 2 langages 254
condition nécessaire 15-17, 30, 339
condition suffisante 15-17, 339
conditionnelle de McCarthy 99
cône de descendance 171
confluence de traitements 195, 213
congruence modulo k 59, 72, 74, 89
conjonction 3, 5, 8, 10, 12, 14-16, 27, 302-303, 313, 339
connecteur 3, 12-14
connexité 171, 183-185, 191, 192
forte 184, 185, 196
consensus 237, **290-292**, 294, 301, 303
conséquent 4, 13
contradiction 6, 9-11, 14-17, 339
contraposition 6, 10, 11, 16, 248
contrôle sémantique des programmes .. 260
contrôles 237
corde d'un chemin 139
corps **265**, 268
de Galois 265, 310
fini 265
correspondance univoque 51
courants fictifs de Maxwell 195
couverture 29, 305
finie 30
irredondante .. 30, 33, 293-295, 298, 299
optimale 30, 295
croisement virtuel 313
cryotrons 284
cycle 184
eulérien 191
hamiltonien 191
sans circuit 184

D

dates
au plus tard 166, 275
au plus tôt 166, 275
décidabilité 95
décision multicritères 80-83
décomposition canonique d'une relation n-aire 60
déduction 5, 7-11, 14-15
définition récursive 100
convergente 101
divergente 101, 102
stationnaire 102
degré d'un sommet (graphe) 130
demi-degré
extérieur *Voir* sortance
intérieur *Voir* entrance
demi-treillis 78, 89, 253, 256, 298

démonstration....4, 7-11, 14, 15
automatique.... 209, 300
locale....5, 7
par l'absurde....6
déparenthèsage....231-234
dérécursivation.... 121, 327
dérivation au sens d'une grammaire.....254
dérivée au sens de Booth....328
dérivée d'une expression régulière 328, 329
descendance **171**, 172
descendant....140
détachement 5, 8, 9, 14, 15, 339
développement....236
diagramme
de décision binaire 307
de Veitch-Karnaugh.... **289**, 294, 299
de Venn....24
d'Euler 22
en losange....71, 79
sagittal 42-43, 67, 257
diamètre d'un graphe....186
dictionnaires multilingues 64, 65
différence ensembliste....28, 29
différence symétrique de 2 ensembles...31, 32
dimension....259
d'une relation n-aire.... 59, 216
dioïde.....263, 264, 266-270, 273, 275, 277
finiment engendré 263
idempotent....264
simplifiable....264
dipôle....283
diséquations....211
disjonction *Voir* adjonction
distance....62
à vol d'oiseau 62
dans R^k....62
de Hamming....63, 167, 217, 286
de Manhattan....62
discrète....62
entre relations 180
euclidienne 62
ultramétrique 85, 88
urbanistique....62
distributivité 236, 301
mutuelle.... 28, 282
divergence 101, 195, 213
divisibilité.... **67**, 68, 71, 76, 78
documentation automatique
bruit....272
champ sémantique....272
évocation 272, 273
silence....272
domination 80, 81
double dualisation....304
double implication....3, 5, 16
double négation 6
duale d'une fonction booléenne.... 300
dualité 14, 283
d'hypergraphes 215

E

écart entre deux sommets.... 185
écartement d'un sommet.... 186
échelle....76, 202
élément
absorbant.... 229
d'ordre n dans un groupe 250
finalement périodique 246
idempotent 230
involutif.... 229
k-périodique.... 246
majorant 77
maximal 77
minimal.... 77
minorant.... 77
neutre 229, 247
régulier *Voir* simplifiable
simplifiable.... 230, 248
éléments
équivalents.... 72
remarquables.... **228-230**, 240, 263
emploi du temps 64, 210
endomorphisme 238
énoncé.... 3, 13
composé....3, 12, 13
élémentaire....3, 4, 11-13
universellement vrai 8
énonciation de chemins.... 153, 159, 265
ensemble **21**
d'articulation.... **190**, 216
d'arcs.... 68, **129**
d'attributs.... 61
de cardinalité n 21, 58
de formes normales 239
de générateurs....251, 259, 263, 322
de points.... 68, **129**
de procédures....164, 165, 179
de renseignement.... 278
de sommets 68, **129**, 214, 216
de tâches 151, 166
de valuation 135, 265
défini en compréhension 22
défini en extension 21
dénombrable 58, 251
des applications de A dans B 53
des booléens.... 23, 282
des complexes.... 23
des entiers naturels 23, **104**
des entiers signés.... 23, **104**
des états internes.... 174, 317

des fonctions f de X à Y 89
des fractions 105
des isométries.....258
des morphismes.....238
des nombres premiers 17
des parties d'un ensemble..... 25, 53, 78 214
des positions non perdantes.....204
des prédécesseurs 130, 140
des rationnels..... 23, 105
des réels 23, 106
des règles d'une grammaire..... 254, 321
des relations de A à B 44, 47
des relations réflexives 78
des relations sur un ensemble.....78
des rotations 249, 258
des séquences bâties sur A..... 251, 252
des successeurs..... 130, 140
des translations..... 249, 258
fini..... 33, 55, **58**
inclus au sens large 23
inclus au sens strict 24
optimal au sens de Paretto 81
récursif.....103
récursivement énumérable.....103
redondant..... **30**, 293
référentiel 24
symétrisé d'un autre ensemble 104
vide 22
ensemble-quotient
attaché à un préordre..... 83, 173
attaché à une équivalence 74
ensembles
de même cardinalité.....21, **55**
disjoints 27
identiques 21
remarquables 23
entier quadrat frei.....89
entrance **130**, 169, 195
épimorphisme.....238
équation
booléenne 308, 309, 312
dans un dioïde 322-326
dans un groupe 249
linguistique 322-326
équité d'un tournoi.....192
équivalence 72
associée à un préordre.....83
associée à une relation 172
attachée à une application.....74
attachée à une partition 74
d'automates 318
logique 5, 21, 180, 312
espace des critères.....81
essais et erreurs 158, 159
estimation de la complexité..... 112-114
estimation de la durée..... 113, 118
d'accès 122
estimation des besoins en mémoire..... 114, 118, 122
étape..... 212
état stable..... 319
évaluation
de droite à gauche..... 231
de gauche à droite..... 231
d'une expression postfixée..... 235
partielle d'une fonction 62, 100
évocation 272, 273
évolution à long terme 177
existence de chemin 148, 153, 159
de taille k 145
expansion de Kleene 323
expression.....224, 225
conditionnelle 99
postfixée..... 233
postfixée au plus tard 234
postfixée au plus tôt 234
préfixée 232
régulière 324
sémantiquement bien formée..... 225
syntaxiquement bien formée..... 224

F

$\mathbb{F}$ *Voir* ensemble des fractions
factorisation..... 236
faisceau de chemins 153, **266**, 272
fermeture réflexo-transitive 171-174, 177-180
fermeture transitive 171, 179
feuilles 118, **187**
figure de Kuratowski..... 131
finesse
entre couvertures 29
entre partitions..... 34
entre relations 46
flèche 68, **129**
fonction
antitone 75
blindée..... 103
calculable.....95, 100, 102
composée 99
de A sur B 54
de Fibonacci.....16, 117-121
de Grundy **202**, 203-208
de marquage d'arcs 265
de transition 317
d'intérêt..... 82
élémentaire..... 98
en programmation 52
évaluation partielle 62, 100
généralisée 61
inversible 54

isotone 75
monotone....75
non calculable 95
ordinale (pour un graphe) 164, 166, 187, 188
partielle....53
partiellement calculable....95
plafond / plancher 52
projection....98
pure 52
récursive 100
récursivement calculable 96
semi-calculable....103
successeur.... 98, 104
surjective 53, 54
fonction booléenne
aiguillage 284
auto-duale **300**, 313
ceinture du cube 287
comme produit de sommes....300
comme somme de produits....288
de 2 variables....286
duale 300
duplex 284, 305
incomplète 297-299
première forme canonique 287
seconde forme canonique 301
synthèse arborescente 306
très incomplète 305
troisième forme canonique 311
fonctionnement global 174
fonctions booléennes duales
par paires 300, 313
force d'évocation 272, 273
forêt 187
formalisation
des énoncés....12
des propositions....3
des termes 12
forme canonique 239, 240, 277
première forme booléenne 287
seconde forme booléenne 301
troisième forme booléenne 311
forme close....330
forme normale....239
fraction 105
irréductible 105
réduite....105
frange....144, 145, 155-157
fraternité universelle 86, 277

G

générateur.....250, 251, 263, 321, 322, 331
gerbier **264**, 270
absorbant 268
fini....268
germe
d'un langage 255
d'une démonstration.... 4, 7, 9
d'une théorie 226
grafcet....**212**-214
grammaire....254-256
ambiguë.... 255, 330
de classe 2.... 327
de classe 3.... 321
de Kleene 321
formelle....254-256
linéaire droite.... 321
linéaire gauche.... 321
linéaire unilatère.... 321
graphe 42, 67, **129**-144
biparti.... 202, **212**, 215
canonique.... 307
complémentaire 133
complet 131
connexe 184, 185
d'appel inter-procédures .. 164, 165, 196
d'implication 180, 358
de structure 195, 196
de transition 175-177, **320**, 322, 328-331
fortement connexe.... 184, 185
inverse.... 134
irréflexif.... 133, 202
k-arête-connexe 191, 192
k-connexe.... 191, 192
marqué 265
multifonctionnel 307
multiparti 201
multivalué....273, 284
orienté 129, 183
partiel.... 131, 189
p-chromatique....207-211
réflexif.... 133
sans circuit..... 151, 152, 154, 187, 188, 204, 267, 356
signé....178, 277, 278
squelette.... 71
symétrique 69, **134**, 183, 184
topologique.... 266
transitif.... 134, 139
valué....135, 186, 265, 266
V-réflexif.... 269
graphe-quotient 141, 142, 168, 173, 176, 177
groupe....**249**-251, 256-259, 264
à 4 éléments....250, 251
abélien ou commutatif.... 251
additif.... 249
cyclique d'ordre n.... 250
de Klein....250, 251
des isométries 258
des permutations.... 257
des signatures dimensionnelles.... 259

monogène 250
multiplicatif 250

H

heuristiques 116, 160
hiérarchie.....101, **187**, 188, 196, 203
homomorphisme *Voir* morphisme
hyperarête.....214-217
hypergraphe.....214-216, 219
hyperonyme.....272
hyponyme.....272
hypothèse 6-11, 15, 16, 339, 340
du monde clos 312

I

idempotence 26, 27, 147, **228**, 245, 264
282, 322
identité de deux ensembles.....21
impasse..... 255, 264, 265
implication ...3-15, 180, 303, 313, 339-341
impossibilité
dans une équation booléenne... 308, 309
inclusion large.....23, 24
inclusion stricte.....24, 25
indétermination
dans une équation booléenne... 308, 309
indicateurs de robustesse.....192
indice
de complexité 113, 114, 150
de coût 150, 288, 292
indifférence 81-83
inf-demi-treillis **78**, 89, 253, 256, 298
injection..... 54, 56, 64, 347
interclassement.....124
interdits élémentaires.....303
interprétation d'une algèbre formelle....226
intersection
d'ensembles.....27
de relations **46**, 47, 78
intransitivité 15, 70, 84, 85
involution 6, 11, 14, 45, 227, 283
d'un opérateur unaire.....227
irréflexivité..... 15, 69, 84, 85
isomorphisme..... 238, 243
isotone 75
isovalence.....74, 81
isthme **191**, 192

J

jalousie universelle 86, 277
jeux sur un graphe.....204-207
jointure de relations n-aires.....61

L

labyrinthe.....139, 155, 158, 159
langage.....254-256
ambigu 255
de classe 2..... 38
engendré par une grammaire..... 255
formel..... 254
liste d'adjacence..... 132
liste réduite 81
logique
des prédicats 11-15
des propositions.....3-11, 312
d'ordre 0 *Voir* des propositions
d'ordre 1 *Voir* des prédicats
longueur
du plus court chemin.....62, 270
d'un chemin 150, 155

M

machine de Turing 96-98, 107
majorant d'un sous-ensemble..... 77
majority gate *Voir* opérateur booléen
maj
marge 275
marquage d'arcs 265
marque d'absence..... 265, 266
matrice
caractéristique d'une relation..... 41, 67,
278
d'adjacence.....131, 132, 192
d'énonciation 136, 192
des débits 135, 276
des distances.....135, 270, 274
diagonale par blocs.....72, 73, 172
d'incidence..... 132, 192
maximum d'un ensemble ordonné..... 77
membres d'un ensemble **21**, 37, 201
métaphore du labyrinthe 139
métathéorème 7
méthode des consensus 290-292, 301
minimum d'un ensemble ordonné..... 77
minorant d'un sous-ensemble..... 77
modèle
de Mealy 317
de Moore..... 318
de tautologie 7-9
d'une algèbre formelle 226
linguistique 254-256, 317, 321
modélisation 69, 93, 193, 254, 283
monoïde **245**, 246-249, 251-253
additif..... 246
commutatif..... 263
de composition 245, 247
finiment engendré..... 251
monôme canonique 287-290, 306, 307

monôme premier ... 289, 290, 293, 294, 304
inutile ... 293
obligatoire ... 293
redondant ... 299
utile ... 293
monotonie ... 75
morphisme ... **237**, 238
bijectif ... 238
injectif ... 238
subjectif ... 238
multigraphe ... **192**, 215, 320
multiplicateur binaire ... 314
complet ... 314

N

ℕ *Voir* ensemble des entiers naturels
négation ... 3, 6, 9-12, 14, 15
involutive ... 10
neurones ... 297
noeuds ... **187**, 194-196
nombre
algébrique ... 106
chromatique ... 207-209
cyclomatique ... 193-196
de connexité ... 185
de Fibonacci ... 16, 117-121
de termes *Voir* arrité
entier naturel ... 104
entier signé ... 104
premier ... 17, 78, **106-107**
quadrat-frei ... 89
rationnel ... 105
réel ... 105
transcendant ... 105
virgule flottante ... 53, 106
non-contradiction ... 3, 10, 23
non-déterminisme ... 331
non-réflexivité ... 69
non-symétrie ... 69
non-transitivité ... 70
notation
infixée ... 224
polonaise *Voir* préfixée
polonaise inverse *Voir* postfixée
postfixée ... 224, 233-236
préfixée ... 224, 232
puissance ... 246
RPN ... 233
notion ... 254, 321, 322, 329, 330
noyau d'un graphe ... 204, 206
n-uples ... 58, 59, 273, 274

O

opérande ... 225
opérateur ... **223-225**, 240, 245, 263
absorbant ... **236**, 282, 293, 310, 322, 325
associatif ... 26, 27, 46, **227**, 231, 240, 245, 282, 310
binaire ... 223, 224
commutatif ... 26, 27, **227**, 240, 248, 251 263
distributif ... 28, 46, **236**, 263, 282, 310, 322
externe ... 223
fondamental ... 240
hétérogène ... 224
idempotent ... 26, 27, **228**, 245, 264, 282 322
interne ... 223
involutif ... 227, 283
secondaire ... 240
simplifiable ... 228, 248, 310
stable ... 224
ternaire ... 223, 224
unaire ... 223, 224
opérateur booléen
aiguillage ... 284
duplex ... 284
et ... 282, 297, 310
maj(orité) ... 313
nand *Voir* on
ni ... 296, 297
non ... 282-286, 296, 300-305, 309-313
on ... 286, **296**
ou ... 46, 47, 147-149, 227, 228, 236, 263, **282**-288, 296, 297, 300-305, 308-313
ou exclusif ... 310-313
optimisation progressive ... 161
orbe d'une permutation ... 164
ordonnancement ... 137, 138, 151, 164, 166-167
lexicographique ... 240
ordre
d'inclusion ... **23-25**, 78
d'un groupe fini ... 250
de contenance ... 264
de divisibilité ... **67**, 68, 71, 76, 78
de finesse ... 29-30, 33, 34, 46, 71, 78
de précédence ... 86, 151, 253
de préfixation ... **246**, 253, 282
de réalisabilité ... 298
du dictionnaire *Voir* ordre lexicographique
entre ensembles ... 23-25
induit sur des tuples ... **76**, 77, 80
induit sur l'ensemble-quotient attaché à un préordre ... 83, 84, 173-175
large ... 23, **75**, 85, 246, 253, 298
lexicographique ... **77**, 86, 253
linéaire ... 76
définition ... 75

partiel 75, 79, 253, 291, 298
strict 25, **76**, 85, 164, 246, 289
total **75-77**, 253
unanime **76-77**, 80
organigramme 325
orthant 77
outil logique universel 313
outils à fenêtre.....314
outils à seuil **297**, 314

P

paires ordonnées 34
palindrome 324
paquet de règles 312, 322
parenthèse.....225, 227, 231, 232, 234, 322
partition 32-34
attachée à une équivalence 73, 74
discrète.....32, 34
d'un ensemble 32, 33
universelle 32, 34
pavé numérique.....259
permutation 55, 164, 257
p-graphe *Voir* multigraphe
physique qualitative 178, 278
piège dans un jeu 206, 207
pilotage
d'une recherche en profondeur 160, 161
plage d'arrêt 101, 102
plan d'action.....139
point..... 22, 42, 68, 128, 129, 287-295
critique..... 31, 295
d'articulation 128, **190**, 191
fixe d'un opérateur..... 228, 239, 324
politiques.....82
posynômes.....115
précédence entre tâches 86, 151, 164
précédence lexicographique.....213
prédécesseur.....130
au rang k.....140
prédicat..... 11-15, 22, 41, 44, 95, 107
atomique 15
calculable.....95
partiellement calculable.....96
préfixe commun 246
le plus long 253
préordre 79, 80, 85, 172
de complexité 115
de domination.....80
de préfixation 245
préordre induit
par une fonction 80
par une notation.....80
préordre le plus fin majorant
une relation 172
preuves d'un calcul.....237
priorité entre opérateurs 232
problème 139
aux dioïdes.....263, 271
des 4 couleurs 208, 209
des ponts de Königsberg 163
du facteur chinois 163
du représentant de commerce,
du VRP 162
eulérien 162
hamiltonien 161
processus de Markoff..... 177, 180
production..... 254
ambiguë..... 255
linguistique 254
terminale vide 255
produit cartésien
de 2 ensembles..... 35
généralisé 59
produit terme à terme..... 47
produit de clauses.....300-304
profil
d'évaluation 59, 61, 80, 84, 91
d'un opérateur..... 223
profondeur de récursivité..... 114, 118, 119
programme structuré 325-327
projection d'une relation n-aire 60
proposition.....13, 302, 303, 313
propriété..... 13
caractéristique d'un ensemble..... 25
faible / forte 135, 184, 185
générique 25
spécifique..... 25
typique 26
pseudo-associativité 231, 232
puissance du continu 106

Q

$\mathbb{Q}$ *Voir* ensemble des rationnels
quadrant.....
quadrat-frei 89
quadripôle..... 283
quantificateur..... 12-15
$\forall$ 12-15
$\exists$ 12-15
$\exists!$ 92, 239, **352**
quasi synonyme..... 272
question
décidable..... 95
semi-décidable..... 96
questionnaire 187

R

$\mathbb{R}$ Voir ensemble des réels
racine

BeatriceGloria_personal library

d'une arborescence 187, 307
n-ième de l'unité 250
raisonnement spiral 215
rayon d'un graphe 186
recherche de chemin
best-first *Voir* le meilleur d'abord
breadth-first *Voir* par niveaux
dans un labyrinthe 139, 155, 158
de taille bornée 153, 160
de taille minimale 153, 160
depth-first *Voir* en profondeur
en profondeur 158-161
le meilleur d'abord 155-157
méthodes globales 143
méthodes matricielles 145-152, 263 265-274
optimal 271
par amélioration progressive ... 160, 161
par niveaux 152-155
problèmes-types 143
procédé Dijkstra-Moore *Voir* le meilleur d'abord
récurrence 14, 93, 101
récursion terminale 327
récursivité
convergente 101
de surface.....327
divergente 101
et récurrence 101
intrinsèque 327
plage d'arrêt 101, 102
profondeur 114, 118, 119
reflet
d'une expression régulière..... 324, 325
d'une séquence 324
et parenthèses 324
réflexivité **69**, 71, 72, 75, 79
réfutation 6, 9-11, 14, 15
règle d'inférence 4-7, 9-11, 13, 14, 339
aristotélicienne 6
du syllogisme 8, 9
positive 5
secondaire 7
sur la négation 6
règle de De Morgan 10, 11
règle de grammaire 254, 255
linéaire unilatère.....321, 322, 329-331
relation.. 12, 41, 47, 59-61, 67-84, 171-174
relation binaire 41-47
comme transformation 44
composée 46, 47
diagramme sagittal 42, 43
domaine 43
en compréhension 44
en extension 41
fonctionnelle 51
image ou codomaine 43
inverse 45
liste de coupes 42
nulle ou vide 44
préfixe 245, 246, 253, 282
régulière 51
totale 52, 53
universelle ou complète 44
vecteur de vérité 42
relation binaire interne 41, 67, 129, 133
antisymétrique 69, 84
asymétrique 69, 84
catégories 85
circulaire 86
d'équivalence **72**, 85
d'identité **68**, 72, 173
d'inclusion **23-25**, 78
d'ordre large .. 23, **75**, 85, 246, 253, 298
d'ordre lexicographique **77**, 86, 253
d'ordre linéaire 76
d'ordre partiel 75, 79, 253, 291, 298
d'ordre strict .. 25, **76**, 85, 164, 246, 289
d'ordre unanime **76-77**, 80
de congruence modulo k 72, 74, 89
de contenance 264
de divisibilité **67**, 68, 71, 76, 78
de finesse 29-30, 33-34, 46, 71, 78
de précédence 86, 151, 253
de préfixation **246**, 253, 282
de préordre 79, **80**, 85
de réalisabilité 298
intransitive 15, 70, 84
irréflexive 15, 69, 84
matrice caractéristique 67
non réflexive 69
non symétrique 69
non transitive 70
nulle ou vide **68**, 85
propriétés 84
réflexive . 69, 71, 84, 133, 146, 147, 171
restriction 72
symétrique 69, 71, 84, 133
symétrisée 71, 135
transitive **70**-72, 75, 76, 84, 85, 171, 172
universelle ou complète . **68**, 71, 72, 173
V-réflexive **269**, 270
relation n-aire 59, 61,
décomposition canonique 60
dimension **59**, 60, 216
projection sur la i-ème dimension 60
relation quaternaire 59
relation quinaire 60
relation ternaire 59
relations binaires
composition .. **46**, 47, 57, 146, 147, 171, 247
finesse **46**, 75, 57, 146, 147, 171, 247
intersection **46**, 47, 78

union **46**, 69, 78, 146, 171
représentant
d'une classe d'équivalence **74**, 238, 239
représentation géométrique
des fonctions booléennes **287**, 292
représentations parfaites 238
représentations simplifiées 237
réseau de communication 185, 187, 192, 196
réseau de contacts 283, 284
réseau de Petri 214
résolution de problèmes..... 139, 187
restriction
d'une relation binaire interne..... 72, 131
restructuration de programmes.....326
rotations..... 226, 257, 258

S

schéma de programme 326
schéma de relations.....61
semi-anneau **264**, 279
semi-calculabilité 96, **103**
semi-groupe..... **248**, 252, 264
commutatif 248
signature
dimensionnelle 259, 260
d'un opérateur **223**, 225
simplifiabilité 228, 248, 249, 252, 264
simplification d'énoncé 8
solutions dominantes 81, 82
solutions dominées 81, 82
sommet
initial..... **130**, 187, 188, 199
isolé..... 68, **130**, 173
pendant 199, 208
terminal..... **130**, 187-189, 202, 204, 206
sommets adjacents **130**, 183
sorites 9, 17, 180, 340
sortance **130**, 132, 154, 159, 169, 195
sortes.....225
sous-ensemble.....**23**, 29, 130, 131, 249 293, 298, 299
fortement connexe maximal *Voir* composante fortement connexe
strict 24
sous-graphe 130
complet *Voir* clique
sous-groupe **249**, 250
squelette d'un graphe transitif..... 71, 134
stabilité d'un opérateur 104, 105, **224**, 249, 264, 325
structure d'influence 178, 277, 278
substitué *Voir* substitution
substitution 4, 7-9, 14, 15, 62, 305
successeur 130, 226
au rang k 140
suite de Fibonacci.....16, 117-121
suite finie 58, 59
suite indexée 58
sup-demi-treillis 78
surcharge d'un opérateur 225
surclassement *Voir* domination
surjection 53-57, 75, 347
syllogisme..... 8, 9, 17, 180, 340, 358
syllogismes en chaîne *Voir* sorites
symétrie
d'une relation binaire interne..... 69
symétrique
dans un monoïde.....247, 249
symétrisé
d'un graphe..... 135
symétrisée
d'une relation binaire interne..... **71**, 135
synonymie 272
système linéaire..... 194, 269
systèmes à états 139, 174-178, 317-321, 331-333

T

$\mathbb{T}$ .... *Voir* fonction booléenne incomplète
table d'adjacence..... 132
table des distances *Voir* matrice des distances
taille
d'un chemin **139**, 145-148
d'un circuit..... 141
tautologie 7-9
temps.....92, 319
terme 11-14, 226
à terme 47, 273
clos 12, 14, 226
transversal 291
théorie 93, 94, 137, **226**
thèse de Church 96
tiers exclus..... 10, 11
tournoi.....162, **168**, 192
tracé d'un seul trait..... 163
tracé point par point 258, 259, 277
traducteur..... 321
transition 139, 175-177, 212-214, 317-320, 322, 328-
transitivité..... 70, 75, 76, 84, 85, 171, 172
traveling salesman problem 162
treillis **78**, 79, 89
d'inclusion des parties d'un ensemble 78
de divisibilité 78
de finesse entre relations binaires 46, **78**

tri 86, 123, **349**
tripôle 285
triples 59

U

union
de relations **46**, 69, 78, 146, 171
d'ensembles 26
univers du discours **24**, 28, 29

V

valuation d'un faisceau de chemins 266
variable 225, 226
booléenne 308, 309, 312
linguistique 323-326
propositionnelle 3, 4, 10, 302
vecteur *Voir* n-uple homogène
booléen 42, 59
de notation *Voir* profil d'évaluation
de vérité 285
vivariance 59
vocable 272
vocabulaire 254, 321
des graphes 199
des notions 254, 321
terminal 254, 321
vote-taker *Voir* opérateur booléen maj
V-réflexivité 269, 270

Z

$\mathbb{Z}$ *Voir* ensemble des entiers signés
zone d'influence 172

par
Achevé d'imprimer
en janvier 2002
IMPRIMERIE LIENHART
à Aubenas d'Ardèche

Dépôt légal janvier 2002
N° d'imprimeur : 3952
Printed in France